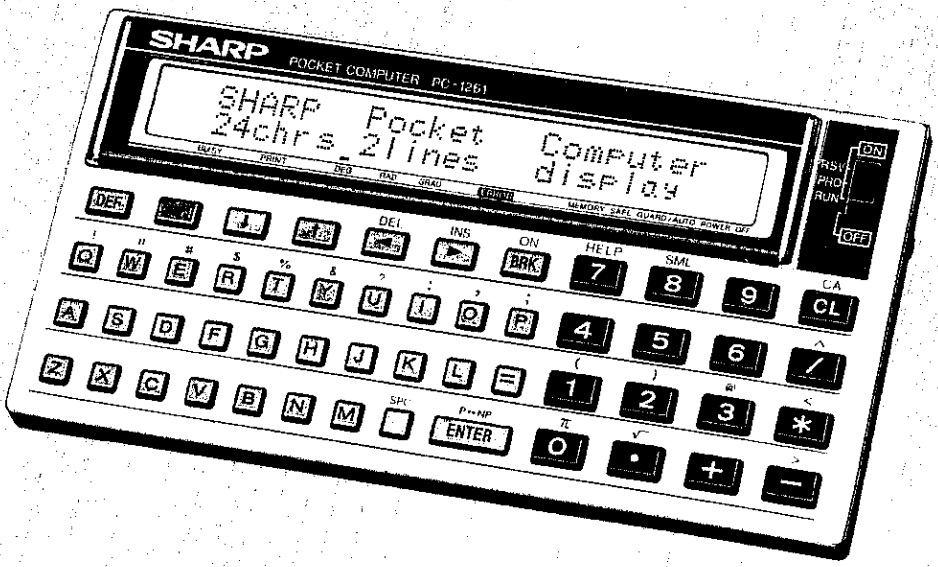
SHARP

TASCHENCOMPUTER

MODELL **PC-1260**
PC-1261

BEDIENUNGSANLEITUNG



S H A R P

T A S C H E N C O M P U T E R

P C - 1 2 6 0 / 1 2 6 1

BEDIENUNGSANLEITUNG

INHALTSVERZEICHNIS

EINLEITUNG		1
KAPITEL 1	WIE MAN DIESES HANDBUCH BENUTZT	3
KAPITEL 2	EINFÜHRUNG IN DEN PC 1260/61	5
KAPITEL 3	RECHNEN OHNE PROGRAMMUNTERSTÜTZUNG	13
KAPITEL 4	BEGRIFFE UND AUSDRÜCKE DES BASIC	32
KAPITEL 5	PROGRAMMIEREN MIT DEM PC 1260/61	47
KAPITEL 6	ABKÜRZUNGEN	56
KAPITEL 7	DER GEBRAUCH DES DRUCKER/MIKROCASSETTEN-RECORDERS CE-125 UND ANDERER OPTIONEN	60
	Der Gebrauch der Option CE-125 (Drucker-Mikrocassettenrecorder)	60
	Der Gebrauch des CE-126P Drucker/ Cassettenrecorder-Interface	73
KAPITEL 8	BASIC-REFERENZ-TEIL	81
	Kommandos	83
	Befehle	96
	Funktionen	144
	Pseudovariable	144
	Numerische Funktionen	147
	String-Funktionen	151
	Kommandofunktionen	153
KAPITEL 9	EASY SIMULATION PROGRAM (ESP)	159
KAPITEL 10	HELP-FUNKTIONEN	172
	Referenz-Funktion	172
	ASCII-Tabelle	174
	Fehlermeldungsfunktion	176
KAPITEL 11	PANNENHILFE	178
	Bedienung des Geräts	178
	Fehlersuche in BASIC	179

KAPITEL 12	DIE INSTANDHALTUNG DES PC 1260/61	181
PROGRAMMBEISPIELE		182
ANHANG A	FEHLERMELDUNGEN	A-1
ANHANG B	ASCII-CODE TABELLE	B-1
ANHANG C	FORMATIEREN DER DATENAUSGABE	C-1
ANHANG D	BEWERTUNG VON AUSDRÜCKEN UND OPERATOREN-VORRANG	D-1
ANHANG E	TASTENFUNKTIONEN	E-1
ANHANG F	TECHNISCHE DATEN	F-1
ANHANG G	DIE BENUTZUNG VON PROGRAMMEN, DIE FÜR DEN PC 1245 ODER PC 1250/51 ENTWICKELT WURDEN	G-1
	ÄNDERUNGEN BEZÜGLICH DES PC 1401	G-4

EINLEITUNG

WILLKOMMEN IN DER WELT DER SHARP-BESITZER!

Nur wenige Industriezweige der Welt können heute Schritt halten mit dem schnellen Wachstum und dem technischen Fortschritt im Bereich der Personalcomputer. Computer, die noch vor kurzer Zeit einen ganzen Saal gefüllt hätten, deren Programmierung einen akademischen Abschluß erfordert und die Tausende von Dollars gekostet hätten, passen heute in Ihre Handfläche, sind einfach zu programmieren und kosten so wenig, daß fast jeder sie sich leisten kann.

Ihr neuer SHARP PC 1260/1261 wurde entwickelt, um Ihnen den allerneuesten Stand dieser Computerrevolution zugänglich zu machen. Als einer der der raffiniertesten Taschencomputer unserer Zeit verfügt er über eine Vielzahl fortschrittlichster Fähigkeiten:

- SPEICHERSCHUTZ - der PC 1260/1261 "merkt sich" gespeicherte Programme und Variablen, selbst wenn Sie ihn abschalten.

- Stromversorgung durch Batterien: ideal für alle, die ein wirklich portables System benötigen.

- AUTOMATISCHE ABSCHALTUNG, die die Batterien schützt, indem die Stromversorgung abgeschaltet wird, wenn innerhalb eines bestimmten Zeitraums keine Operation durchgeführt wird.

- Programmierbare Funktionen, mit deren Hilfe Sie Ihren PC 1260/1261 als "klugen" Rechner benutzen können.

- Eine erweiterte Version des BASIC, die formatierte Datenausgabe ermöglicht, weiterhin zweidimensionale Felder, variable String-Länge, Programmverkettung und viele andere fortschrittlichste Arbeitsmöglichkeiten.

- EASY SIMULATION PROGRAM - diese Funktion wird benutzt, um einzelne Werte in einer laufenden Programmierung schnell auszuwechseln.

- HELP-FUNKTION: wird benutzt, wenn Sie die Erklärung eines Fehlers benötigen, oder wenn Sie lediglich die Anzeige von Kontrolldaten und Operationsvorgängen wünschen.

  (1) Die Referenz-Funktion zeigt BASIC-Befehle und ihre richtige Anwendung.

  (2) Die ASCII-Table-Funktion zeigt ASCII-Codes in hexadezimaler Schreibweise.

  (3) Die Fehlermeldungs-Funktion erklärte den jeweiligen Fehlertyp und zeigt ihn mit Hilfe des Cursors im Zeilendisplay an.

- Als Option: Drucker/Mikrocassettenrecorder (Modell CE-125) für Langzeitspeicherung und Ausdruck von Programmen und Daten als Hard-Copy.

Wir gratulieren Ihnen zu Ihrem Eintritt in eine aufregende und faszinierende neue Welt. Wir sind sicher, daß Sie Ihre neue Errungenschaft schon bald als eine der klügsten betrachten werden, die Sie je getätigt haben. Der SHARP PC 1260/1261 ist ein hilfreiches Werkzeug, das entwickelt wurde, um Ihren spezifischen mathematischen, wissenschaftlichen, technischen, kaufmännischen und persönlichen Computerbedürfnissen gerecht zu werden. Mit dem SHARP PC 1260/61 können Sie JETZT damit anfangen, die Lösungen zu finden, die Sie morgen brauchen werden.

KAPITEL 1

WIE MAN DIESES HANDBUCH BENUTZT

Dieses Handbuch soll Sie einführen in die Fähigkeiten und Charakteristika Ihres PC 1260/1261 und es soll Ihnen als wertvolles Nachschlagewerk dienen. Ob Sie ein "Einsteiger" sind oder ob die Computer zu Ihren "alten Bekannten" zählen: Sie sollten sich mit dem PC 1260/1261 vertraut machen, indem Sie die Kapitel 2 bis 6 lesen und durcharbeiten.

- Kapitel 2 beschreibt das äußere Erscheinungsbild Ihres PC 1260/1261.

- Kapitel 3 erklärt den Gebrauch des PC 1260/1261 als normalen Taschenrechner.

- Kapitel 4 erklärt einige Ausdrücke und Begriffe, die für die Programmierung in BASIC von Bedeutung sind, und es erläutert Ihnen die spezielle Bedeutung dieser Begriffe für die Arbeit mit dem PC 1260/1261.

- Kapitel 5 dient als Einführung in die BASIC-Programmierung auf dem PC 1260/1261. Es zeigt, wie man Programme eingibt, korrigiert und abarbeiten läßt.

- Kapitel 6 schließlich behandelt einige Tips und Kniffe, mit denen Sie sich die Arbeit mit Ihrem PC 1260/1261 noch weiter vereinfachen und noch mehr Spaß damit haben können.

Erfahrene BASIC-Programmierer können dann noch Kapitel 8 durchlesen, um die spezifischen Züge des BASIC im PC 1260/1261 kennenzulernen. Da sich alle BASIC-Dialekte geringfügig unterscheiden, empfiehlt es sich, dieses Kapitel mindestens einmal durchzulesen, bevor man mit dem ernsthaften Programmieren beginnt.

Kapitel 8 ist ein Nachschlageteil, der alle Befehle, Kommandos und Funktionen des BASIC in übersichtlicher, alphabetisch gruppierter Anordnung enthält.

Wenn Sie bislang noch nicht in BASIC programmiert haben, schlagen wir vor, daß Sie sich hierfür ein separates Lehrbuch zulegen oder an einem BASIC-Kurs teilnehmen, ehe Sie diese Kapitel durcharbeiten. Dieses Handbuch ist nicht als Lehrbuch für das Programmieren konzipiert.

Der übrige Teil des Handbuchs besteht aus:

- Kapitel 7 - Grundlegende Informationen über die Option CE-125, Drucker/Mikrocassettenrecorder sowie andere Zusatzgeräte.
- Kapitel 9 - Erklärungen zum Gebrauch des Easy Simulation Programms.
- Kapitel 10 - Erklärung der Help-Funktion.
- Kapitel 11 - Ein Leitfaden, der Ihnen helfen soll, Probleme der Bedienung und Programmierung zu lösen.
- Kapitel 12 - Pflege und Wartung Ihres neuen Computers.

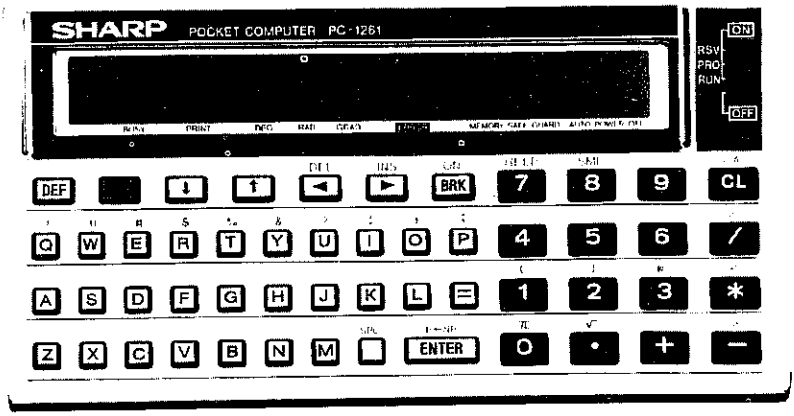
Ein ausführlicher Anhang am Ende dieses Handbuches stellt Ihnen nützliche Tabellen, Vergleichsübersichten und einige weitere Hinweise zum Gebrauch und zur Funktionsweise Ihres PC 1260/1261 zur Verfügung.

KAPITEL 2

EINFÜHRUNG IN DEN PC1260/1261

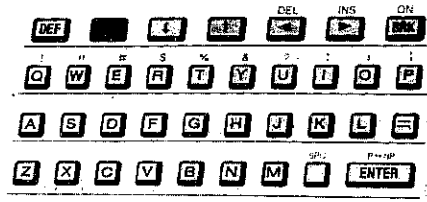
Das System SHARP PC 1260/1261 besteht aus:

- 52-Zeichen-Tastatur
- 24-Zeichen-Anzeige, zweizeilig
- 8-Bit CMOS Prozessor
- 40 KB ROM
- 4.4 KB RAM (PC 1260) bzw. 10.4 KB RAM (PC 1261)
- Option: CE-125 Drucker/Mikrocassettenrecorder



Um Sie mit der Anordnung und den Funktionen von Teilen der Tastatur Ihres PC 1260/1261 vertraut zu machen, wollen wir nun die einzelnen Bestandteile der Tastatur genauer betrachten. Für den Anfang sollten Sie zunächst die Positionen der einzelnen Tasten feststellen und die dazugehörigen Beschreibungen lesen. In Kapitel 3 werden wir anfangen, mit unserem neuen Gerät zu arbeiten.

BESCHREIBUNG DER TASTATUR



[A] - [Z] Alphabetische Tastatur. Sie kennen diese wahrscheinlich von einer normalen Schreibmaschine. Wenn Sie diese Tasten betätigen, werden Großbuchstaben angezeigt. Werden die SHIFT - und SML -Taste nacheinander gedrückt, erscheint "SMALL" links in der Anzeige. Wird nun ein Buchstabe eingegeben, erscheint er als Kleinbuchstabe. Wenn Sie SHIFT und SML noch einmal nacheinander drücken, verschwindet die "SMALL"-Anzeige und es werden wieder Großbuchstaben angezeigt.

[=] Gleichheitszeichen. Auf dem PC 1260/1261 wird diese Taste nicht benutzt, um das Ende einer Rechenoperation anzuzeigen; dieses Symbol hat eine spezielle Funktion in der BASIC-Programmierung.

[SPC] Leertaste. Das Betätigen dieser Taste befördert den Cursor um einen Schritt nach rechts, wobei ein Leerfeld hinterlassen wird. Betätigt man SPC , während der Cursor auf einem Buchstaben steht, wird dieser Buchstabe mit einem Leerfeld überschrieben.

P↔NP
[ENTER] ENTER -Taste. Wenn man diese Taste betätigt, wird, was immer vorher eingegeben wurde, in den Speicher des Computers befördert. Die Funktion ähnelt u. a. der Wagenrücklauftaste auf einer Schreibmaschine. Man muß ENTER drücken, bevor der PC 1260/1261 auf alphanumerische Eingaben über die Tastatur reagiert.
Drückt man SHIFT vor Betätigung von ENTER -Taste, bewirkt das, daß manuelle Rechenoperationen auf dem CE-125-Drucker ausgedruckt werden. Mit der gleichen Tastenfolge (SHIFT ENTER) wird die Option wieder ausgeschaltet.

[DEF] Diese spezielle Taste wird zur Abarbeitung von BASIC-Programmen benutzt.

[SHIFT] Doppelfunktionstaste. Wenn Sie diese Taste vor einer anderen Taste betätigen, wird das über der Taste stehende Zeichen angezeigt.

[↓] Pfeil-nach-unten-Taste. Diese Taste bewirkt, daß die nächste Zeile im Display erscheint.

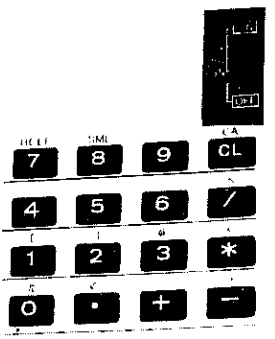
[↑] Pfeil-nach-oben-Taste. Diese Taste bewirkt, daß die vorherige Zeile ins Display gerufen wird.

DEL [◄] Rückschritt-Taste. Mit Hilfe dieser Taste können Sie den Cursor nach links bewegen, ohne dabei vorher eingegebene Zeichen zu löschen. Betätigen Sie [SHIFT] , ehe Sie diese Taste drücken, wird das Zeichen, auf dem der Cursor jeweils steht, gelöscht.

INS [►] Vorwärts-Taste. Diese Taste ermöglicht die Bewegung des Cursors nach rechts, ohne daß vorher eingegebene Zeichen gelöscht werden. Im [SHIFT]-Modus setzt diese Taste einen Leerschritt unmittelbar vor das Zeichen, auf dem der Cursor jeweils steht, so daß Sie hier eine Einfügung machen können.

ON [BRK] BREAK -Taste. Die [BRK]-Taste wird zur Unterbrechung des Programmlaufs benutzt. Nach Abschalten des Rechners durch die Automatik wird er mit dieser Taste wieder eingeschaltet.

! " #
$ % &
? : ,
; Diese Symbole befinden sich über der oberen Reihe der Alpha-Tastatur. Betätigt man [SHIFT] und dann den Buchstaben unter dem entsprechenden Zeichen, erscheinen diese Symbole in der Anzeige.



ON
RSV
PRO
RUN
OFF Schiebeschalter zum Ein- und Ausschalten des Rechners Beachten Sie, daß das Gerät eingeschaltet ist, wenn der Schalter in einer der drei Positionen RUN, PROgramm oder ReSerVe steht!

[Ø]-[9] [.] Zifferntasten. Die Anordnung dieser Tasten ähnelt der auf einem gewöhnlichen Taschenrechner.

SML
[8] Wenn [SHIFT] und [SML] nacheinander gedrückt werden, erscheint "SMALL" in der Anzeige und Kleinbuchstaben werden angezeigt. Werden [SHIFT] und [SML] noch einmal nacheinander betätigt, wird der Kleinschriftmodus wieder aufgehoben.

HELP
[7] Die HELP-Funktion wird durch das Betätigen von [SHIFT] und [HELP] aufgerufen. Durch Druck auf die [CL]-Taste lassen Sie den PC 1260/1261 in den vorherigen Modus zurückkehren.

CA
[CL] Durch das Betätigen der CLEAR -Taste werden alle alle eingegebenen Zeichen gelöscht und die Blockade durch Fehlermeldung aufgehoben. Drückt man [SHIFT] , bevor man diese Taste betätigt, so wird die [CA] (=Reset)-Funktion aktiviert. [CA] löscht die Anzeige und initialisiert den Computer.

[/] Divisionszeichen. Mit dem Schrägstrich gibt man dem PC 1260/1261 zu verstehen, daß man eine Division ausführen möchte. Im [SHIFT]-Modus fungiert es als Exponentiations-Zeichen, das dem Computer anzeigt, daß eine Zahl potenziert werden soll.

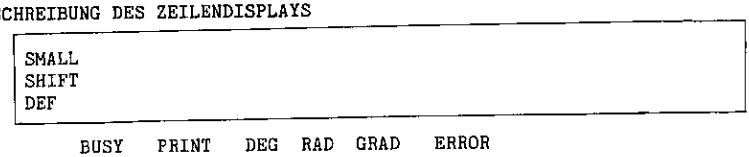
<
[*] Multiplikationszeichen. Der Stern steht beim PC 1260/1261 wie bei den meisten anderen Computern für die Multiplikation. Im [SHIFT]-Modus dient es zur Darstellung der Relation "kleiner als".

>
[-] Subtraktionszeichen. Stellt im [SHIFT] -Modus die Relation "größer als" dar.

[+] Additionszeichen.

π √ @
( ) Diese Symbole befinden sich über der Null, dem Dezimalpunkt und den Ziffern 1, 2 und 3. Sie werden angesprochen, indem man vor Betätigung der entsprechenden Taste [SHIFT] drückt.

BESCHREIBUNG DES ZEILENDISPLAYS



Die Flüssigkristallanzeige des SHARP PC 1260/1261 zeigt bis zu 48 Zeichen (einschließlich des Cursors) zur Zeit. Obwohl man bis zu 80 Zeichen (einschließlich [ENTER]) in einer Zeile eingeben kann, werden nur die ersten 48 Zeichen angezeigt. Um die übrigen Zeichen der Zeile sichtbar zu machen, muß man die [↓]-Taste drücken, dann wird die Anzeige "nach oben rollen".

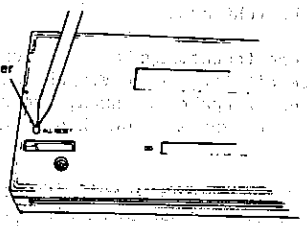
Das Anzeigefeld besteht aus:

>	Bereitschaftssymbol. Es erscheint, wenn der Computer bereit ist, eine Eingabe aufzunehmen. Sobald Sie mit der Eingabe beginnen, verschwindet das Bereitschaftssymbol und wird durch den Cursor ersetzt.
_	Cursor. Dieses Symbol (Unterstrich) markiert die Stelle Stelle des nächsten einzugebenden Zeichens. Wenn man mit der Eingabe von Daten beginnt, ersetzt der Cursor das Bereitschaftssymbol. Als Markierungssymbol wird der Cursor auch im Zusammenhang mit den INSert- und DELete-Funktionen benutzt.
SMALL	Anzeige für Kleinschriftmodus. "SMALL" erscheint in in der Anzeige, wenn [SHIFT] und [SML] nacheinander gedrückt werden. Bei Anzeige von "SMALL" werden bei Betätigung der Alphatastatur Kleinbuchstaben angezeigt. Der Kleinschriftmodus wird durch erneutes Betätigen von [SHIFT] und [SML] aufgehoben, der Computer kehrt in den Großschriftmodus zurück.
SHIFT	Anzeige der Doppelfunktion. Der Schriftzug leuchtet auf, wenn die Doppelfunktionstaste betätigt wurde. Bitte beachten Sie, daß die [SHIFT] -Taste wieder losgelassen werden muß, bevor eine andere Taste betätigt wird.
DEF	Define-Modus. Diese Anzeige leuchtet auf, wenn die [DEF] -Taste betätigt wurde.
- BUSY	Programmausführungsanzeige. Sie erscheint, während der PC 1260/1261 ein Programm abarbeitet oder eine Berechnung durchführt (außer, wenn Zeichen ausgegeben werden). Die Abschalteautomatik wird ausgesetzt, wenn BUSY angezeigt ist. Die Anzeige verschwindet, wenn die Ausführung des Programms beendet ist.

- PRINT — Druckeranzeige. Dieses Symbol erscheint, wenn beim Rechnen ohne Programmunterstützung der Drucker angewählt wurde oder wenn das Easy Simulation Programm benutzt wird.
Wenn [SHIFT] und [P•NP] nacheinander gedrückt werden, erscheint das PRINT-Symbol. Es verschwindet bei wiederholtem Betätigen von [SHIFT] und [P•NP].

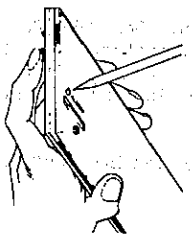
- DEG RAD GRAD — Anzeige der Winkeleinheit.

- ERROR — Fehleranzeige. Erscheint immer dann, wenn ein Fehler auftritt. Die Blockade durch den Fehler beseitigen Sie durch Betätigen der [CL] -Taste.



ALL RESET: Reset-Schalter. Er wird gebraucht, um den Computer zu initialisieren, wenn CLear oder CA nicht ausreichen, um das Problem zu lösen.

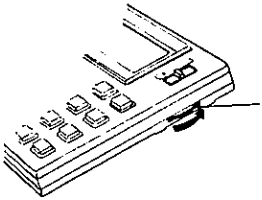
ACHTUNG: Um den PC 1260/1261 zu initialisieren, drücken Sie irgendeine Taste, halten diese fest und drücken gleichzeitig den ALL-RESET-Schalter auf der Rückseite des Gerätes. Die Programme, Variablen und der Reservespeicher werden dabei nicht gelöscht.



Zum Drücken des All-Reset-Schalters benutzen Sie bitte einen spitzen Gegenstand, z. B. einen Kugelschreiber. Vermeiden Sie Gegenstände, deren Spitzen abbrechen könnten (z. B. Bleistifte!).

Sollte Ihr PC 1260/1261 nach dieser Operation noch nicht reagieren, betätigen Sie den All-Reset-Schalter, ohne dabei eine eine andere Taste zu drücken. Dabei werden allerdings Speicherinhalte, Daten und Programme gelöscht - man sollte also den All-Reset-Schalter allein nur betätigen, wenn es unumgänglich ist.

KONTRASTKNOPF



Kontrastknopf: Durch Drehen des Kontrastknopfes in Pfeilrichtung wird der Kontrast der Anzeige verstärkt, in entgegengesetzter Richtung verringert.

BATTERIEWECHSEL

Der PC 1260/1261 arbeitet ausschließlich mit Lithium-Batterien. Wenn er an die Option CE-125 angeschlossen ist, kann der PC 1260/1261 auch über diese versorgt werden, sofern deren momentane Betriebsspannung höher ist. Dies verringert den Stromverbrauch der Lithium-Zellen.

Beim Wechseln der Batterien beachten Sie bitte unbedingt die folgenden Hinweise:

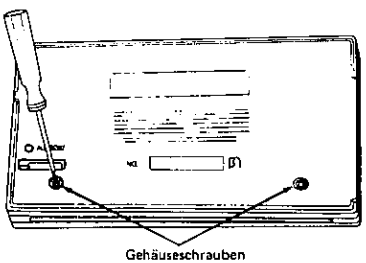
- Wechseln Sie grundsätzlich beide Batterien gleichzeitig.
- Verwenden Sie niemals eine neue gemeinsam mit einer gebrauchten Batterie.
- Benutzen Sie nur Lithium-Zellen (Typ CR-2032).

EINSETZEN DER BATTERIEN

Wenn die Anzeige bei maximal eingestelltem Kontrast schwach und schlecht zu erkennen ist, zeigt dies an, daß die Batterien verbraucht sind. In diesem Fall müssen Sie die Batterien umgehend erneuern. (Wenn Sie die Option CE-125 benutzen, sollten Sie Programme und Daten vorher auf Cassette abspeichern.)

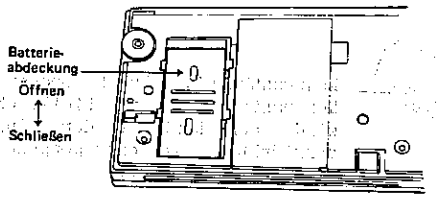
(1) Schalten Sie den Computer ab, indem Sie den Schiebeschalter in die OFF-Position bringen.

(2) Entfernen Sie die Schrauben auf dem rückwärtigen Gehäusedeckel mit einem kleinen Schraubendreher.

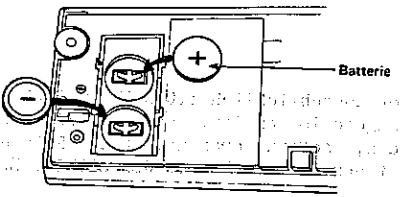


Gehäuseschrauben

(3) Schieben Sie die Batterie-Abdeckung in Pfeilrichtung, sie läßt sich dann abheben.

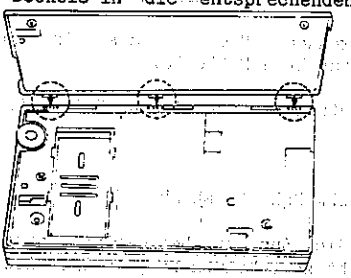


(4) Erneuern Sie beide Batterien.



(5) Setzen Sie die Batterie-Abdeckung durch Schieben gegen die Pfeilrichtung wieder ein.

(6) Setzen Sie die Klauen des Deckels in die entsprechenden Aussparungen am PC 1260/1261.



(7) Klappen Sie den Deckel zu und befestigen Sie ihn mit den Gehäuseschrauben.

(8) Bringen Sie den Schiebeschalter in die Stellung ON und betätigen Sie den RESET-Schalter, um den Computer zu initialisieren. In der Anzeige sollte stehen:

>

Wenn die Anzeige leer ist oder ein anderes Zeichen als das Bereitschaftssymbol erscheint, entfernen Sie die Batterien und setzen Sie sie erneut ein, kontrollieren Sie dann wieder das Display.

Achtung: Läßt man eine leere Batterie im Computer, kann er durch Säureaustritt aus der Batterie geschädigt werden. Leere Batterien umgehend erneuern!

Vorsicht:	Bewahren Sie die Batterien außerhalb der Reichweite von Kindern auf!

KAPITEL 3

RECHNEN OHNE PROGRAMMUNTERSTÜTZUNG

Nachdem Sie nun vertraut sind mit den Bestandteilen Ihres SHARP PC 1260/1261 und deren Anordnung, wollen wir damit beginnen, die faszinierenden Fähigkeiten Ihres neuen Computers zu entdecken.

Weil der PC 1260/1261 über alle gebräuchlichen mathematischen Funktionen verfügt und darüber hinaus in BASIC programmierbar ist (nützlich für komplexere Rechenoperationen), wird er häufig als "schlauer" Rechner bezeichnet. Das macht natürlich aus Ihnen einen "schlauen" Benutzer.

(Bevor Sie nun mit Ihrem PC 1260/1261 arbeiten, vergewissern Sie sich, daß die Batterien korrekt eingesetzt sind.)

EINSCHALTEN DES COMPUTERS

Um den PC 1260/1261 einzuschalten, betätigen Sie den Schiebeschalter und wählen Sie einen der drei Modi RUN, PRO oder RSV. Um den PC 1260/1261 wie einen einfachen Taschenrechner zu benutzen, muß der RUN-Modus gewählt werden. Wenn das Gerät eingeschaltet ist, erscheint das Bereitschaftssymbol (>) in der Anzeige.

AUSSCHALTEN DES COMPUTERS

Um den PC 1260/1261 auszuschalten, bringen Sie den Schiebeschalter in die Position OFF.

Wenn Sie das Gerät auf "OFF" schalten, löschen Sie das Zeilendisplay. Die Programme und Daten bleiben jedoch erhalten und stehen zur Verfügung, sobald das Gerät wieder eingeschaltet wird.

Wenn die BEEP-Anweisung oder der CLOAD-Befehl ausgeführt werden, unterbrechen Sie die Ausführung, indem Sie die [BRK] -Taste drücken und den Schiebeschalter in die Position OFF bewegen.

AUTOMATISCHE ABSCHALTUNG

Um die Batterien vor unnützer Beanspruchung zu schützen, schaltet sich der PC 1260/1261 automatisch ab, wenn 11 Minuten lang keine Taste betätigt wurde. (Achtung: Die automatische Abschaltung wird ausgesetzt, solange der PC 1260/1261 ein Programm abarbeitet!)

Alle Daten und Programme bleiben bei automatischer Abschaltung gespeichert. Wenn Sie den Computer nach automatischer Abschaltung wieder starten wollen, tun Sie dies mit der [ON/BRK] -Taste.

EINIGE HILFREICHE HINWEISE

Bis Sie sich an Ihr neues Gerät gewöhnt haben, werden Sie sicherlich Fehler bei der Dateneingabe machen. Wir werden später einige einfache Korrekturmöglichkeiten kennenlernen. Für's erste: Wenn Sie eine Fehlermeldung erhalten, drücken Sie CLear und wiederholen die Eingabe. Wenn der Computer "sich aufhängt" - d.h. überhaupt nicht mehr reagiert - betätigen Sie den ALL-RESET-Schalter (siehe Kapitel 2).

Das Bereitschaftssymbol (>) sagt Ihnen, daß der PC 1260/1261 eine Eingabe erwartet. Wenn Sie Daten eingeben, verschwindet das Bereitschaftssymbol und der Cursor bewegt sich nach rechts und zeigt dabei die nächstverfügbare Stelle im Zeilendisplay an.

Mit den Pfeilen nach rechts (►) und links (◄) bewegen Sie den Cursor innerhalb einer Zeile.

[ENTER] informiert den PC 1260/1261, daß Sie die Dateneingabe beendet haben und fordert den Computer auf, die gewünschten Operationen durchzuführen. SIE MÜSSEN [ENTER] AM ENDE JEDER EINGABEZEILE BETÄTIGEN; ANDERNFALLS WERDEN IHRE RECHNUNGEN VOM COMPUTER NICHT AUSGEFÜHRT WERDEN!

Wenn Sie numerische Berechnungen durchführen, erscheint die Eingabe links in der oberen Zeile des Displays, das Ergebnis steht rechts in der unteren Zeile.

Benutzen Sie die [SHIFT] -Funktion in Verbindung mit einer anderen Taste (um z. B. die Quadratwurzel zu ziehen), drücken Sie [SHIFT], lassen Sie [SHIFT] wieder los und drücken dann die andere Taste. Der [SHIFT] -Modus bezieht sich lediglich auf die nächste gedrückte Taste.

Benutzen Sie keine $-Zeichen oder Kommata, wenn Sie mit dem PC 1260/1261 ohne Programmunterstützung rechnen. Diese Zeichen haben eine besondere Bedeutung in der Programmiersprache BASIC.

In diesem Handbuch benutzen wir die Ø, um die Null zu kennzeichnen, so daß zwischen der Ziffer (Ø) und dem Buchstaben (O) unterschieden werden kann.

Um Ihnen die korrekte Dateneingabe am Anfang zu erleichtern, werden wir in den Beispielrechnungen jede nötige Tastenbedienung zeigen. Wenn die [SHIFT] -Funktion angesprochen wird, werden wir das gewünschte Zeichen im folgenden Tastensymbol abbilden. So entsteht beispielsweise durch Betätigen von [SHIFT] und [2] das "="-Zeichen. Eine solche Tastenbedienung wird mit [SHIFT] [=] dargestellt.

Denken Sie daran, nach jeder Rechenoperation die CLear-Taste zu betätigen. (Es sei denn, Sie möchten Kettenrechnungen durchführen.) CLear löscht die Anzeige und beseitigt eventuelle Blockaden durch Fehler. Speicherinhalte werden nicht gelöscht.

EINFACHE RECHENOPERATIONEN

Der PC 1260/1261 rechnet auf 10 Stellen genau. Wenn Sie es nicht bereits getan haben, schalten Sie jetzt Ihren Computer ein, indem Sie den Schiebeschalter auf RUN stellen. Nun versuchen Sie, die folgenden einfachen Rechenbeispiele nachzuvollziehen. Denken Sie daran, zwischen einzelnen Berechnungen CLear zu betätigen.

EINGABE	ANZEIGE
5 0 + 5 0 ENTER	50 + 50 100.
1 0 0 - 5 0 ENTER	100 - 50 50.
6 0 * 1 0 ENTER	60 * 10 600.
3 0 0 / 5 ENTER	300 / 5 60.
1 0 SHIFT ^ 2 ENTER	10 ^ 2 100.
2 * SHIFT π ENTER	2 * π 6.283185307
SHIFT √ 6 4 ENTER	√64 8.
4 E 3 ENTER	4 E 3 4000.

WIEDERABRUF VON EINGABEN

Auch nachdem der PC 1260/1261 die Ergebnisse Ihrer Rechenoperationen angezeigt hat, können Sie Ihre vorherige Eingabe wieder abrufen. Dazu benutzen Sie die Rechts- ▶ und Links- ◀ -Pfeile.

Der Links-◀-Pfeil ruft den vorherigen Ausdruck ab, wobei der Cursor hinter dem letzten Zeichen steht.

Der Rechts-▶-Pfeil ruft den vorherigen Ausdruck ab, jedoch steht der Cursor in diesem Fall auf dem ersten Zeichen.

Sie erinnern sich: der Rechts- und Links-Pfeil wird auch benutzt, um den Cursor auf der Zeile zu bewegen. Die Pfeile sind sehr hilfreich beim Kontrollieren oder Verändern von Eingaben, ohne dabei den gesamten Ausdruck noch einmal eingeben zu müssen.

Sie werden mit dem Gebrauch der Rechts- und Links-Pfeile im Verlauf der folgenden Beispiele schnell vertraut werden. Schlüpfen Sie nun in die Rolle eines Managers und führen unsere Beispielrechnungen selbst durch, während wir sie erklären.

Als Personalchef in einem großen Marketing-Unternehmen sind Sie verantwortlich für die Planung der jährlichen Verkaufstagung. Sie erwarten zu diesem dreitägigen Treffen 300 Personen. Für einen Teil dieser Zeit wird man sich in kleine Arbeitsgruppen aufteilen. Sie sind der Ansicht, daß Gruppen zu je 6 Personen eine gute Größe wären. Wieviele Gruppen müssen Sie einplanen?

EINGABE	ANZEIGE
3 0 0 / 6 ENTER	300 / 6 50.

Dann fällt Ihnen ein, daß Gruppen mit einer ungeraden Teilnehmerzahl möglicherweise effektiver arbeiten. Rufen Sie Ihre letzte Eingabe mit Hilfe des ◄-Pfeils wieder auf.

EINGABE	ANZEIGE
◄	300 / 6_

Um die neue Anzahl der Gruppen zu berechnen, müssen Sie die 6 durch eine ungerade Zahl ersetzen. Fünf Gruppenmitglieder scheinen Ihnen sinnvoller als sieben. Da Sie daran gedacht haben, den ◄-Pfeil zu benutzen, steht der Cursor nun am Ende der Anzeige. Benutzen Sie noch einmal den ◄-Pfeil, um den Cursor eine Stelle nach links zu bewegen.

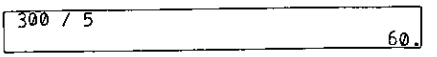
EINGABE	ANZEIGE
◄	300 / 6

Beachten Sie, daß nach dieser Bewegung der Cursor sich in ein blinkendes Viereck verwandelt. Immer, wenn Sie den Cursor auf ein schon bestehendes Zeichen setzen, wird er in dieser Form dargestellt.

Geben Sie nun eine 5 anstelle der 6 ein. Achtung beim Ersetzen von Zeichen: Wenn Sie ein neues Zeichen über ein bereits bestehendes setzen, ist das ursprüngliche endgültig gelöscht; Sie können keinen Ausdruck wieder abrufen, der überschrieben wurde.

EINGABE ANZEIGE

[5] [ENTER]

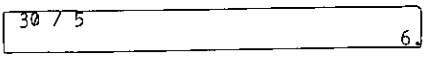


Sechzig scheint Ihnen eine vernünftige Anzahl von Gruppen zu sein, so beschließen Sie, daß jede Kleingruppe aus 5 Teilnehmern bestehen wird.

Erinnern Sie sich, daß es sinnvoll ist, die letzte Eingabe zu kontrollieren, besonders, wenn Ihre Ergebnisse Ihnen zweifelhaft erscheinen. Nehmen Sie z. B. an, Sie hätten folgende Rechnung durchgeführt:

EINGABE ANZEIGE

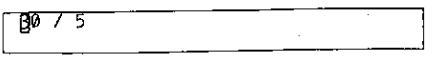
[3] [0] [/] [5] [ENTER]



Selbst ein müder, überarbeiteter Manager, wie Sie es sind, bemerkt, daß 6 kein glaubwürdiges Ergebnis ist, wenn man mit Hunderten von Personen operiert. Rufen Sie Ihre Eingabe mit Hilfe des [►] -Pfeiles noch einmal ins Zeilendisplay.

EINGABE ANZEIGE

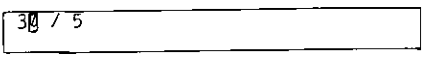
[►]



Weil Sie den [►] -Pfeil benutzt haben, steht der blinkende Cursor jetzt auf dem ersten Zeichen der Anzeige. Um die Eingabe zu korrigieren, möchten Sie jetzt eine zusätzliche Null einfügen. Mit Hilfe des [►] -Pfeils bewegen Sie den Cursor, bis er über der Null steht. Wenn Sie nun mit der [INS] -Taste eine Einfügung machen wollen, setzen Sie den blinkenden Cursor auf denjenigen Buchstaben, vor dem das neue Zeichen eingefügt werden soll.

EINGABE ANZEIGE

[►]



Benutzen Sie die [INS] -Taste, um Platz für das benötigte Zeichen zu schaffen.

EINGABE ANZEIGE

[SHIFT] [INS]

Das Betätigen der [INS] -Taste bewegt alle Zeichen um einen Schritt nach rechts und fügt einen von Klammern eingeschlossenen Leerraum ein. Der blinkende Cursor steht nun auf diesem Leerraum und zeigt die Stelle der nächsten Eingabe an. Geben Sie die fehlende Null ein. Nachdem die Eingabe korrigiert ist, rufen Sie erneut Ihr Ergebnis ab.

EINGABE	ANZEIGE
[0] [ENTER]	300 / 5 60.

Stellen Sie sich auf der anderen Seite vor, Sie hätten die folgende Eingabe gemacht:

EINGABE	ANZEIGE
[3] [0] [0] [0] [/] [5] [ENTER]	3000 / 5 600.

Das Ergebnis scheint viel zu groß. Wenn Sie nur 300 Personen zu Ihrem Treffen erwarten, wie könnte es dann 600 Arbeitsgruppen geben? Rufen Sie Ihre Eingabe mit Hilfe des [►]-Pfeils ab.

EINGABE	ANZEIGE
[►]	3000 / 5

Der blinkende Cursor steht nun über dem ersten Zeichen der Anzeige. Um die Eingabe zu korrigieren, müssen Sie eine der Nullen löschen. Mit Hilfe des [►] -Pfeils bewegen Sie den Cursor auf die erste (oder eine andere) Null. Wenn Sie ein Zeichen löschen wollen, setzen Sie den Cursor auf das zu löschende Zeichen.

EINGABE	ANZEIGE
[►]	3000 / 5

Nun betätigen Sie die [DEL] -Taste, um eine der Nullen zu beseitigen.

EINGABE	ANZEIGE
[SHIFT] [DEL]	300 / 5

Durch Betätigen der [DEL] -Taste rutschen alle Zeichen um einen Schritt nach links. Das Zeichen, auf dem der Cursor steht, wird gelöscht, ebenso der Freiraum, den dieses Zeichen beanspruchte. Der blinkende Cursor bleibt an derselben Stelle stehen und zeigt die nächste Stelle für eine Eingabe ein. Da Sie keine weiteren Veränderungen vorzunehmen haben, beenden Sie die Rechnung.

EINGABE	ANZEIGE
[ENTER]	300 / 5 60.

(Achtung: Steht der Cursor auf einem Zeichen und Sie betätigen die [SPC] -Taste, wird das betreffende Zeichen gelöscht und durch einen Leerschritt ersetzt. Mit [DEL] löschen Sie das Zeichen und den von ihm eingenommenen Leerraum.)

FEHLER

Der Abruf der vorherigen Eingabe ist notwendig, wenn Sie die gefürchtete ERROR-Meldung erhalten. Wir wollen uns vorstellen, daß Sie unabsichtlich folgendes in den PC 1260/1261 eingegeben haben:

EINGABE	ANZEIGE
[3][Ø][Ø][/][/][5][ENTER]	ERROR 1

Natürlich sind Sie überrascht, wenn diese Meldung erscheint! ERROR 1 ist einfach die Art Ihres Computers, Ihnen mitzuteilen: "Ich weiß nicht, was ich hier tun soll."

Wenn Sie in diesem Fall den [◄] -Pfeil betätigen, erscheint der blinkende Cursor an der Stelle, wo der Fehler liegt. Der Fehlertyp wird in der darunterliegenden Zeile angezeigt.

EINGABE	ANZEIGE
[◄]	300 / Ø 5 SYNTAX ERROR

Um diesen Fehler zu korrigieren, benutzen Sie die [DEL] -Taste.

EINGABE	ANZEIGE
[SHIFT] [DEL]	300 / 5

Wenn Sie beim Abruf Ihrer Eingabe aufgrund der Meldung ERROR 1 feststellen, daß Sie ein Zeichen vergessen haben, benutzen Sie die [INS] -Sequenz, um es zu korrigieren.

Wenn Sie den PC 1260/1261 wie einen normalen Taschenrechner benutzen, werden Sie in der Hauptsache mit dem ERROR 1 (Syntaxfehler) konfrontiert werden. Eine komplette Aufstellung der Fehlermeldungen und ihrer Bedeutungen finden Sie in Anhang 1.

VERKETTUNG VON RECHENOPERATIONEN

Ihr PC 1260/1261 erlaubt, daß Sie das Ergebnis einer Rechnung in der folgenden Operation weiterverwenden.

Zurück zu unserem Beispiel: Ein Teil Ihrer Verantwortung bei der Planung dieser Konferenz besteht darin, einen detaillierten Budget-Vorschlag auszuarbeiten. Sie wissen, daß Ihnen für jeden Teilnehmer DM 150,-- zur Verfügung stehen. Berechnen Sie Ihr Gesamtbudget.

EINGABE	ANZEIGE
[3][0][0][*][1][5][0][ENTER]	300 * 150 45000.

Von dieser Summe beabsichtigen Sie, 15 % für die Abschlußfeier aufzuwenden. Wenn Sie verkettete Rechenoperationen durchführen, ist es nicht nötig, Ihr erstes Ergebnis noch einmal einzugeben, wenn Sie die [CL]-Taste NICHT benutzen. Wie hoch ist Ihr Budget für die Feier?

EINGABE	ANZEIGE
[*][.][1][5][ENTER]	45000. * .15 6750.

Beachten Sie, daß der Computer das Ergebnis Ihrer ersten Berechnung automatisch links oben im Zeilendisplay anzeigt und in die neue Rechnung miteinbezieht, wenn Sie die Befehle für die zweite Berechnung eingeben. In verketteten Rechenoperationen muß die Eingabe mit einem Operationssymbol beginnen. Beenden Sie Ihre Eingabe wie immer mit ENTER.

Achtung: Die [%]-Taste kann nicht in Rechenoperationen benutzt werden. Sie sollte ausschließlich als ein Zeichen benutzt werden.

Beispiel:

EINGABE	ANZEIGE
[4][5][0][0][0][*][1][5][SHIFT][%][ENTER]	ERROR 1

Fahren Sie fort, Ihr Budget zu berechnen. Das Hotel verlangt für die Verpflegung DM 4000,--.

EINGABE	ANZEIGE
[-][4][Ø][Ø][Ø] [ENTER]	6750. - 4000 2750.

Die Dekoration wird DM 1225,-- kosten.

EINGABE	ANZEIGE
[-][1][2][2][5] [ENTER]	2750. - 1225 1525.

Schließlich müssen Sie DM 2200,-- für den Sprecher und für das Rahmenprogramm einkalkulieren.

EINGABE	ANZEIGE
[-][2][2][Ø][Ø] [ENTER]	1525. - 2200 -675.

Nun, Sie werden entweder Ihre Pläne oder Ihre Kostenverteilung ändern müssen!

NEGATIVE ZAHLEN

Da die Abschlußfeier etwas ganz Besonderes sein soll, beschließen Sie, Ihre Pläne nicht zu ändern und das zusätzliche Geld zu investieren. Trotzdem fragen Sie sich, wieviel Prozent des Gesamtbudgets wohl von diesem Posten in Anspruch genommen werden. Zuerst ändern Sie das Vorzeichen Ihres letzten Ergebnisses.

EINGABE	ANZEIGE
[*][-][1]	-675. * - 1 675.

Nun addieren Sie dieses Ergebnis zu Ihrem ursprünglichen Budget für die Feier hinzu.

EINGABE	ANZEIGE
[+][6][7][5][Ø] [ENTER]	675. + 6750 7425.

Teilen Sie Ihr neues Ergebnis durch 45000 und Sie erfahren, wieviel Prozent des Gesamtbudgets diese neue Zahl ausmacht.

EINGABE

[/] [4] [5] [0] [0] [0] [ENTER]

ANZEIGE

```
7425 / 45000
                    0.165
```

Nun gut, Sie beschließen, 16,5 % für die Abschlußfeier aufzuwenden.

KOMPLEXERE RECHENOPERATIONEN UND KLAMMERUNG

Bei der Durchführung der obigen Rechenbeispiele hätten Sie einige dieser Operationen in einem Schritt zusammenfassen können. So hätten Sie etwa die beiden folgenden Berechnungen in eine Zeile schreiben können:

```
675 + 6750 / 45000
```

Solche komplexeren Rechnungen müssen jedoch sehr sorgfältig eingegeben werden.

675 + 6750 / 45000 könnte verstanden werden als

$$\frac{675 + 6750}{45000} \quad \text{oder} \quad 675 + \frac{6750}{45000}$$

Bei der Durchführung komplexer mathematischer Operationen folgt der PC 1260/1261 spezifischen Regeln der impliziten Klammerung und der Rangfolge von mathematischen Operatoren (siehe Anhang D). Um sicherzustellen, daß Ihre Eingaben in der von Ihnen gewünschten Reihenfolge verarbeitet werden, sollten Sie Klammern verwenden.

```
(675 + 6750) / 45000   oder  675 + (6750 / 45000)
```

Wie unterschiedlich sich die Plazierung von Klammern auf das Ergebnis auswirken kann, sehen Sie in den beiden folgenden Beispielen:

EINGABE

[SHIFT] [(] [6] [7] [5] [+] [6] [7] [5] [0] [SHIFT] [)] [/] [4] [5] [0] [0] [0] [ENTER]

ANZEIGE

```
(675 + 6750) / 45000
                    0.165
```

[6] [7] [5] [+] [SHIFT] [(] [6] [7] [5] [0] [/] [4] [5] [0] [0] [0] [SHIFT] [)] [ENTER]

```
675 + (6750 / 45000)
                   675.15
```

DER GEBRAUCH VON VARIABLEN IN RECHENOPERATIONEN

Der PC 1260/1261 kann bis zu 26 einstellige numerische Variablen unter den alphabetischen Zeichen A - Z speichern. Wenn Ihnen der Begriff der Variablen nicht vertraut ist: In Kapitel 4 finden Sie ausführliche Erklärungen. Die Variablenzuweisung geschieht über einen Zuweisungsbefehl:

```
A = 5
B = -2
```

Sie können auch den Wert einer Variablen (rechts) einer anderen (links) zuweisen.

```
C = A + 3
D = E
```

Eine Variable kann in jeder Rechenoperation anstelle einer Zahl benutzt werden.

Nachdem Sie nun Ihre Abschlußfeier geplant haben, fehlen Ihnen noch die übrigen Berechnungen für Ihre Verkaufstagung. Sie möchten den Rest Ihres Budgets ebenfalls prozentual aufteilen. Zuerst müssen Sie herausfinden, wieviel Geld Sie noch zur Verfügung haben. Weisen Sie der Variablen (den verbleibenden Rest) zu.

EINGABE

R = 4 5 0 0 0 -
7 4 2 5 ENTER

ANZEIGE

```
R = 45000 - 7425
                37575.
```

Wenn Sie nun ENTER drücken, führt der PC 1260/1261 die Berechnung durch und zeigt den neuen Wert von R an. Sie können den Wert jeder Variablen zur Anzeige bringen, indem Sie den Buchstaben eingeben, unter dem er gespeichert ist.

EINGABE

R ENTER

ANZEIGE

```
R
                37575.
```

Nun können Sie Rechenoperationen mit Ihren Variablen durchführen. Der Wert von (R) wird sich nicht verändern, bis Sie ihm einen neuen Wert zuweisen.

Sie beabsichtigen, 60 % des verbleibenden Geldes auf die Unterbringung zu verwenden.

EINGABE

R * . 6 0 ENTER

ANZEIGE

```
R * .60
                22545.
```

Weiterhin wollen Sie 25 % des verbleibenden Budgets für die Durchführung von Management-Trainings-Seminaren ausgeben.

EINGABE

[R] [*] [.] [2] [5] [ENTER]

ANZEIGE

```
R * .25
                    9393.75
```

Variable behalten ihren zugewiesenen Wert auch, wenn das Gerät ausgeschaltet wird oder sich automatisch abschaltet.

Sie gehen nur verloren, wenn

- man ihnen einen neuen Wert zuweist,
- man CLEAR [ENTER] (nicht die [CL] -Taste) eingibt,
- man das Gerät mit dem ALL-RESET-Schalter initialisiert,
- die Batterien gewechselt werden.

Es gibt gewisse Grenzen der Variablenzuweisung sowie gewisse Progamm-abläufe, die Variablen verändern können. In Kapitel 4 finden Sie detaillierte Informationen zur Variablenzuweisung, in Kapitel 5 wird der Gebrauch von Variablen bei der Programmierung erörtert.

MEHRERE RECHENSCHRITTE

Mit dem PC 1260/1261 ist es nicht nur möglich, verschiedene Rechenschritte zusammenzufassen, man kann auch einzelne Rechnungen nacheinander durchführen, ohne zwischendurch [ENTER] betätigen zu müssen. Die einzelnen Rechenoperationen werden durch Kommata getrennt. In der Anzeige erscheint nur das Ergebnis der letzten Rechenoperation. (Beachten Sie bitte, daß die maximale Zeilenlänge, die Ihr Computer annimmt, 80 Zeichen einschließlich [ENTER] beträgt.)

Sie fragen sich, wieviel Geld für die Unterbringung zur Verfügung stünde, wenn Sie, wie ursprünglich geplant, nur 15 % für die Abschlußfeier aufwenden würden.

EINGABE

[R] [=] [.] [8] [5] [*] [4] [5] [0] [0]
[0] [SHIFT] [,] [R] [*] [.] [6] [0]

ANZEIGE

```
R = .85 * 45000, R * .60
```

Obwohl der Computer alle eingegebenen Rechenoperationen durchführt, zeigt er nur das letzte Ergebnis an.

EINGABE

[ENTER]

ANZEIGE

```
R = .85 * 45000, R * .60
                     22950.
```

Um herauszufinden, welchen Wert die Variable R in dieser Berechnung hatte, geben Sie R ein:

EINGABE	ANZEIGE
R ENTER	R 38250.

WISSENSCHAFTLICHE NOTATION

Wer mit sehr großen und sehr kleinen Zahlenwerten zu arbeiten hat, benutzt oft ein spezielles Format, die sogenannte exponentielle oder wissenschaftliche Notation. In der wissenschaftlichen Notation wird eine Zahl in zwei Teile aufgespalten.

Der erste Teil besteht aus einer Dezimalzahl zwischen 1 und 0. Der zweite Teil gibt die Größe der Zahl in Zehnerpotenzen an.

Wie Sie wissen, ist in einer gewöhnlichen Dezimalzahl die erste Stelle links vom Komma die Stelle der Einer, an zweiter Stelle folgen die Zehner, an dritter die Hunderter, dann die Tausender usw. Dies sind einfach ansteigende Potenzen von 10.

$10^0 = 1,\ 10^1 = 10,\ 10^2 = 100,\ 10^3 = 1000$ usw.

Die wissenschaftliche Notation bricht also eine Dezimalzahl in zwei Teile auseinander. Ein Teil zeigt, um welche Ziffernfolge es sich handelt, der zweite, wieviele Stellen eine Zahl links oder rechts vom Komma aufweist. Z. B.:

1234 wird 1.234 mal 10^3 (drei Stellen nach rechts)

654321 wird 6.54321 mal 10^5 (fünf Stellen nach rechts)

.000125 wird 1.25 mal 10^{-4} (vier Stellen nach links)

Wissenschaftliche Notation ist hilfreich, um Eingaben zu verkürzen. Eine 1 mit 87 Nullen wäre doch allzu umständlich zu schreiben, obendrein recht unübersichtlich. In wissenschaftlicher Notation würde diese Zahl folgendermaßen aussehen:

$1.0 * 10^{87}$ oder 1.0 E 87

Der PC 1260/1261 benutzt die wissenschaftliche Notation immer dann, wenn Zahlen in dezimaler Schreibweise die Anzeigekapazität übersteigen würden. Dabei wird der Ausdruck "mal 10 hoch" durch den Großbuchstaben E dargestellt.

123456789000 wird angezeigt als 1.23456789 E 12
.000000000001 wird angezeigt als 1 E -12

Wer mit dieser Art der Notation nicht vertraut ist, sollte einmal einige sehr kleine oder sehr große Zahlenwerte eingeben, um zu beobachten, wie diese dargestellt werden.

GRENZEN

Die größte Zahl, mit der der PC 1260/1261 operieren kann, besteht aus 10 Stellen plus 2 Stellen im Exponenten. Mit anderen Worten: die größte Zahl ist

```
9.999999999 E 99 = 99999999990000000000000000000000000
                   00000000000000000000000000000000000
                   00000000000000000000000000000000000
```

und die kleinste Zahl ist

```
9.999999999 E -99 = .00000000000000000000000000000000000
                    00000000000000000000000000000000000
                    00000000000000000000000000000000009
                    99999999
```

WERTE-SCHABLONE

Im Fall der Kettenrechnungen konnten Sie das Ergebnis der Berechnung lediglich als erstes Glied eines folgenden Ausdrucks verwenden. Betrachten Sie dazu das folgende Beispiel:

EINGABE	ANZEIGE
[3] [+] [4] [ENTER]	3 + 4 7.
[*] [5] [ENTER]	7. * 5 35.

Drücken Sie [CL] , dann die [↑]- oder [↓]-Taste. Wenn Sie diese Tasten unmittelbar nach Abschluß der obigen Berechnung bedient haben, sollten Sie jetzt "35" in der Anzeige sehen. Die ausgegebene Zahl ist das Ergebnis Ihrer letzten Berechnung.

Der PC 1260/1261 "merkt sich" das letzte ermittelte Ergebnis, und mit Hilfe der [↑]- oder [↓]-Taste kann es wieder abgerufen werden.

Im Fall der oben beschriebenen Kettenrechnung konnten Sie das Ergebnis der vorherigen Rechenoperation ausschließlich als erstes Glied des folgenden Ausdrucks benutzen. Mit Hilfe der Werte-Schablone jedoch können Sie das Ergebnis der vorherigen Berechnung an jeder beliebigen Stelle in die anschließende Rechenoperation einfügen.

Beispiel:

Benutzen Sie das Ergebnis (6.25) der Rechnung 50 / 8 um zu berechnen:
12 x 5 / 6.25 + 24 * 3 / 6.25 =

EINGABE	ANZEIGE
[5][0][/][8][ENTER]	50 / 8 6.25
[1][2][*][5][/][↑]	12 * 5 / 6.25_
[↑][↑][↓][*][3][/][↓]	12 * 5 / 6.25 + 24 * 3 / 6.25_
[ENTER]	12 * 5 / 6.25 + 24 * 3 / 6.25 21.12
[CL][↓]	21.12_

Wie in diesem Beispiel gezeigt, kann das letzte Ergebnis jederzeit und an jeder beliebigen Stelle abgerufen werden, es wird jedoch ersetzt, sobald eine neue Berechnung ein neues Endergebnis liefert.

Das gespeicherte Ergebnis wird durch Betätigen der [CL] -Taste nicht gelöscht.

* Das letzte Ergebnis kann nicht abgerufen werden, wenn sich der Computer nicht im RUN-Modus befindet, wenn die Programmausführung vorübergehend unterbrochen ist oder wenn der Trace-Modus gewählt wurde.

LÄNGE VON AUSDRÜCKEN

Die Länge von Ausdrücken, die Sie in Ihren Computer eingeben können, unterliegt gewissen Begrenzungen. Mit dem PC 1260/1261 können bis zu 79 Tastenbedienungen für einen einzigen Rechenausdruck verwendet werden (ausgenommen die [ENTER] -Taste). Wenn Sie ein 80. Zeichen eingeben wollen, wird der Cursor (▯) an dieser Stelle zu blinken beginnen und damit anzeigen, daß die 80. Eingabe nicht gewertet wird.

TABELLE FÜR DIE WISSENSCHAFTLICHEN FUNKTIONEN

Der PC 1260/1261 ist mit den in dieser Tabelle angegebenen mathematischen Basisfunktionen ausgestattet. Beachten Sie bitte, daß die Darstellung in BASIC sich von der normalen mathematischen Schreibung unterscheiden kann.

FUNKTION	mathematische Schreibweise	BASIC Befehl	Bemerkungen
trigonometische Funktionen	sin cos tan	SIN COS TAN	
trigonometische Umkehrfunktionen	arcsin arccos arctan	ASN ACS ATN	
10er Logarithmus	log	LOG	Logarithmus zur Basis 10
natürlicher Logarithmus	ln	LN	Logarithmus zur Basis e
Exponentialfunktion	e	EXP	e=2.718281826
Potenzierung	a^b	^	a^b = a hoch b
Quadratwurzel	$\sqrt{}$	$\sqrt{}$ oder SQR	
Umwandlung Dezimal-Grad nach Winkel-Grad (Grad, Minute, Sekunde)		DMS	bei Winkel-Umwandlung darf nicht wie in DEG .5 die 0 wegfallen
Umwandlung Winkel-Grad (Grad, Minute, Sekunde) nach Dezimal-Grad		DEG	
ganze Zahlen		INT	INT(x) ergibt die größte ganze Zahl kleiner oder gleich x

Betrag	|x|	ABS	ABS(x) ergibt den Betrag von x
Vorzeichen		SGN	SGN(x)= 1 f. x>0 =-1 f. x>0 = 0 f. x=0
Pi	π	π oder PI	π=3.141592654

Winkel einheit	Befehl	Anzeige Symbol	Erklärung
Alt-Grad	DEGREE	DEG	Rechter Winkel = 90 Grad
Bogenmaß	RADIAN	RAD	Rechter Winkel =π/2 rad.
Neugrad	GRAD	GRAD	Rechter Winkel = 100

Diese Befehle werden benutzt, um Winkelmaße in Programmen zu spezifizieren. Benutzen Sie diese Befehle zur Übung in den folgenden Rechenbeispielen:

1. sin 30° =

DEGREE [ENTER]
SIN 30 [ENTER]

```
SIN 30
                                0.5
 DEG
```

2. tan $= \frac{\pi}{4}$

RADIAN [ENTER]
TAN (PI/4) [ENTER]

```
TAN (PI / 4)
                                1.
 RAD
```

3. $\cos^{-1}(-0.5)$=

DEGREE [ENTER]
ACS -.5 [ENTER]

```
ACS -.5
                               120.
```

4. log 5 + ln 5 =

LOG 5 + LN 5 ENTER

```
LOG 5 + LN 5
                    2.308407917
```

5. e^{2+3} =

EXP (2 + 3) ENTER

```
EXP (2 + 3)
                    148.4131591
```

6. $4^3 + 6^4$ =

(4^3 + 6^4) ENTER

```
(4^3+6^4)
                    36.87817783
```

7. Wandle 30 Grad 30' in DMS-Darstellung um in Dezimal-Darstellung (DEG).

DEG 30.30 ENTER

```
DEG 30.30
                    30.5
```

8. Wandle 30.755 in Dezimaldarstellung (DEG) um in DMS-Darstellung.

DMS 30.755 ENTER

```
DMS 30.755
                    30.4518
                    (30°,45min,18sec)
```

RANGFOLGE VON OPERATIONEN

Im BASIC-Modus können Sie Ausdrücke in derselben Reihenfolge eingeben, in der sie geschrieben sind. Die interne Rangfolge bei der Berechnung und die Behandlung von Zwischenergebnissen wird vom Computer kontrolliert.

Die interne Rangfolge beim Rechnen ohne Programmunterstützung ist die folgende:

1) Abruf von Variablen oder Pi
2) Funktionen (sin, cos etc.)
3) Exponentiation (^)
4) Vorzeichen (+, -)
5) Multiplikation oder Division (*,/)
6) Addition oder Subtraktion (+, -)
7) Größenvergleich (>, >=, <, <=, <>)
8) Logisches AND, OR

Anmerkungen:

- Werden in einem Ausdruck Klammern verwendet, hat die innerhalb der Klammern angegebene Operation höchste Priorität.

- Zusammengesetzte Funktionen werden von rechts nach links abgearbeitet (sin cos^{-1} 0.6).

- Mehrstufige Exponentation wird von rechts nach links abgearbeitet ($(3^4)^2$ oder 3^4^2).

- Von den obigen Punkten 3) und 4) hat die letzte Eingabe die höhere Priorität.

  z. B. -2 ^ 4 $\longrightarrow$ - (2^4)

  3 ^ -2 $\longrightarrow$ 3^{-2}

KAPITEL 4

BEGRIFFE UND AUSDRÜCKE DES BASIC

In diesem Kapitel wollen wir einige Begriffe und Ausdrücke der Programmiersprache BASIC kennenlernen.

STRING-KONSTANTEN

Der PC 1260/1261 ist in der Lage, außer Zahlen auch Buchstaben und spezielle Symbole in vielfacher Weise zu verarbeiten. Diese Buchstaben, Zahlen und speziellen Symbole werden Zeichen genannt. Folgende Zeichen stehen dem PC 1260/1261 zur Verfügung:

```
1 2 3 4 5 6 7 8 9 0
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
! " # $ % & ( ) * + , - . / : ; < = > ? @ √ π
```

Im BASIC wird eine Gruppe von Zeichen als String bezeichnet. Damit der PC 1260/1261 den Unterschied zwischen einem String und anderen Programmteilen, wie z. B. Befehlen oder Variablen-Bezeichnungen erkennen kann, muß man die Zeichen, die zu einem String gehören, in Anführungszeichen (") einschließen.

Es folgen einige Beispiele für String-Konstanten:

"HALLO"

"TSCHUESS"

"SHARP PC 1260/1261"

Die folgenden Beispiele werden als String-Konstanten nicht akzeptiert:

"COMPUTER	Anführungszeichen am Ende fehlen.
"IST" NICHT"	Anführungszeichen dürfen nicht innerhalb eines Strings benutzt werden.

HEXADEZIMALZAHLEN

Das Dezimalsystem ist nur eines von verschiedenen Zahlensystemen. Ein anderes, dessen Bedeutung im Zusammenhang mit Computern stark angewachsen ist, ist das Hexadezimalsystem. Das Hexadezimalsystem basiert auf der Zahl 16 statt auf der 10. Um hexadezimale Ziffern zu schreiben, benutzt man die Ziffern 0 - 9 sowie 6 weitere "Ziffern": A, B, C, D, E und F. Diese entsprechen den Zahlen 10, 11, 12, 13, 14, 15 und 16. Wenn Ihr PC 1260/1261 eine Zahl als hexadezimal auffassen soll, setzen Sie ein UND-Zeichen (&) vor den betreffenden Ausdruck.

Hexadezimal		Dezimal
&A	=	10
&10	=	16
&100	=	256
&FFFF	=	65535

VARIABLE

Computer sind aufgebaut aus einer Vielzahl von kleinsten Speichereinheiten, genannt Bytes. Jedes Byte kann man sich als einzelnes Zeichen vorstellen. Das Wort "Byte" erfordert beispielsweise vier Speicherbytes, weil es vier Buchstaben enthält. Um herauszufinden, wieviele Bytes zum Arbeiten zur Verfügung stehen, geben Sie einfach MEM [ENTER] ein. Die angezeigte Zahl gibt an, wieviele Bytes zum Programmieren frei sind.

Zahlen werden in codierter Form gespeichert. Aufgrund dieser Codierung ist der Computer in der Lage, auch lange Zahlen in nur 8 Bytes zu speichern. Die größtmögliche Zahl, die gespeichert werden kann, ist +9.999999999 E +99.

Die kleinste speicherbare Zahl ist +1.E-99. So erhalten Sie einen recht großen Zahlenbereich, innerhalb dessen Sie arbeiten können. Wenn jedoch das Ergebnis einer Rechnung diesen Rahmen übersteigt, wird Ihr Computer Ihnen dies mitteilen, indem er eine Fehlermeldung ausgibt (ERROR 2). Eine Erklärung dieser Fehlermeldung finden Sie in Anhang A. Geben Sie nun zur Probe ein.

9E99 * 9 [ENTER]

Damit Ihr Computer wieder normal weiterarbeiten kann, brauchen Sie lediglich die [CL] -Taste zu drücken.

Aber wie speichert man denn nun so viel Information? Das ist ganz einfach! Der Computer zieht es vor, verschiedenen Datengruppen Namen zu geben.

Wir wollen die Zahl 556 abspeichern. Sie können diese Zahl nennen, wie Sie möchten, aber für dieses Beispiel wollen wir ihr den Namen R geben. Die Anweisung LET kann benutzt werden, um den Computer zu veranlassen, einer Variablen einen Wert zuzuweisen - allerdings nur in einem Programmbefehl. Der LET-Befehl ist aber nicht unbedingt erforderlich, darum werden wir ihn nicht sehr oft benutzen. Geben Sie nun R = 556 ein und drücken Sie [ENTER]. Der Computer hat nun den Wert 556 mit dem Buchstaben R in Verbindung gebracht. Diese Buchstaben, die man benutzt, um Information zu speichern, nennt man Variable. Um den Inhalt der Variablen R abzufragen, drücken Sie die [CL] -Taste, dann die Tasten R und [ENTER]. Der Computer antwortet, indem er den Wert 556 rechts unten im Display anzeigt. Diese Möglichkeit kann sehr nützlich sein, wenn man Programme oder Formeln eingibt.

Wir wollen als nächstes die Variable R in einer einfachen Formel verwenden. In dieser Formel steht R für den Radius eines Kreises, dessen Fläche wir berechnen wollen. Die Formel für die Kreisfläche lautet: $A = \pi * R^2$. Geben Sie ein: [R] [SHIFT] [^] [2] [*] [SHIFT] [π] [ENTER]. Das Ergebnis ist 971179.3866. Dieses Verfahren des Gebrauchs von Variablen wird eingehender erklärt, wenn wir uns mit dem Programmieren beschäftigen.

Bislang haben wir uns nur mit numerischen Variablen befaßt. Wie speichert man nun alphabetische Zeichen? Grundsätzlich ist das Prinzip das gleiche, aber damit der Computer den Unterschied zwischen den beiden Variablen-Typen erkennen kann, muß nun ein $ zum Namen der Variablen gesetzt werden. Wir wollen z. B. das Wort BYTE unter der Variablen B$ speichern. Sehen Sie das $-Zeichen hinter dem B? Dies sagt dem Computer, daß der Inhalt der Variablen B alphanumerisch bzw. ein Daten-String ist.

Damit dies klarer wird, geben Sie ein: [B] [SHIFT] [$] [=] [SHIFT] ["] BYTE [SHIFT] ["] [ENTER]. Das Wort BYTE ist nun unter der Variablen B$ gespeichert. Um sich zu vergewissern, geben Sie ein: [B] [SHIFT] [$] [ENTER]. In der Anzeige steht BYTE, diesmal aber auf der linken Seite der Anzeige.

Die Variablen, mit denen der SHARP PC 1260/1261 arbeitet, sind folgendermaßen aufgegliedert:

Variable	numerische Variable	vorgegebene numerische Variable (A bis Z) einfache numerische Variable (AB, C1 usw.) numerische Feld-Variable
	String-Variable	vorgegebene String-Variable (A$ bis Z$), einfache String-Variable (BB$, C2$ usw.), String-Feld-Variable

VORGEGEBENE VARIABLE

Der erste Bereich, die vorgegebenen Variablen, werden vom Computer grundsätzlich benutzt, um Daten zu speichern. Man kann ihn sich als reservierten Bereich denken. Mit anderen Worten: egal, wieviel Speicherplatz Ihr Programm in Anspruch nimmt, Ihnen stehen immer mindestens 26 Variable zur Speicherung von Daten offen. Diese Daten können entweder NUMERISCH oder in STRING (Buchstaben) -Form gewählt werden. Vorgegebene Speicherplätze haben eine Kapazität von 8 Byte und können immer nur einen Datentyp zur Zeit aufnehmen. Geben Sie zur Erklärung ein:

A = 123 [ENTER]

A$ [ENTER]

Sie erhalten die Meldung:

ERROR 9

Das bedeutet, daß Sie numerische Daten in den Speicherbereich mit dem Namen A gelegt haben und daß Sie dann dem Computer befohlen haben, diese Information als String-Daten wieder abzurufen. Das aber verwirrt den Computer, so daß er eine Fehlermeldung ausgibt. Drücken Sie die [CL] -Taste, um die Fehlermeldung zu beseitigen. Versuchen Sie nun das folgende Beispiel:

```
A$ = "ABC" [ENTER]

A [ENTER]
```

Wieder ist der Computer verwirrt und meldet ERROR 9. In der folgenden Abbildung können Sie sehen, daß die Variable A im selben Speicherbereich liegt wie die Variable A$, daß B sich mit B$ deckt usw.

Abbildung:

```
A = A$ = A(1)  = A$(1)
B = B$ = B(2)  = B$(2)
C = C$ = C(3)  = C$(3)
D = D$ = D(4)  = D$(4)
E = E$ = E(5)  = E$(5)
F = F$ = F(6)  = F$(6)
G = G$ = G(7)  = G$(7)
H = H$ = H(8)  = H$(8)
I = I$ = I(9)  = I$(9)
J = J$ = J(10) = J$(10)
K = K$ = K(11) = K$(11)
L = L$ = L(12) = L$(12)
M = M$ = M(13) = M$(13)
N = N$ = N(14) = N$(14)
O = O$ = O(15) = O$(15)
P = P$ = P(16) = P$(16)
Q = Q$ = Q(17) = Q$(17)
R = R$ = R(18) = R$(18)
S = S$ = S(19) = S$(19)
T = T$ = T(20) = T$(20)
U = U$ = U(21) = U$(21)
V = V$ = V(22) = V$(22)
W = W$ = W(23) = W$(23)
X = X$ = X(24) = X$(24)
Y = Y$ = Y(25) = Y$(25)
Z = Z$ = Z(26) = Z$(26)
```

EINFACHE VARIABLE

Einfache Variablenbezeichnungen werden durch zwei (oder mehr) alphanumerische Zeichen charakterisiert, wie z. B. AA oder B1. Anders als feste Variable haben die einfachen Variablen keinen fest im Speicher reservierten Bereich. Der Speicherbereich für einfache Variable wird automatisch im Programm oder im Datenbereich bereitgestellt, sobald eine einfache Variable erstmalig benutzt wird.

Da für einfach numerische und einfache String-Variable verschiedene Speicherbereiche vorgesehen sind, können Variable mit dem gleichen Namen (wie z. B. AB und AB$) gleichzeitig benutzt werden.

Wenn man alphanumerische Bezeichnungen für Variablen benutzen will, muß man als erstes Zeichen grundsätzlich einen Buchstaben wählen. Werden mehr als zwei Buchstaben zur Definition eines Variablen-Namens benutzt, sind nur die beiden ersten Zeichen von Bedeutung.

Achtung:

* Die Namen von Funktionen oder BASIC-Befehlen dürfen auf dem PC 1260/1261 nicht zur Bezeichnung von Variablen benutzt werden (z. B. PI, IF, TO, ON, SIN usw.).

* Unter einer einzelnen String-Variablen können bis zu 16 Buchstaben oder Zeichen gespeichert werden.

FELD-VARIABLE

In einigen Fällen ist es sinnvoll, Zahlen in organisierten Gruppen zu verarbeiten, z. B. eine Tabelle der Fußball-Ergebnisse oder eine Steuertabelle. Im BASIC werden solche Gruppen FELDER genannt. Ein Feld kann eindimensional sein, wie z. B. eine Liste, es kann aber auch zweidimensional sein, wie z. B. eine Tabelle.

Um ein Feld zu definieren, benutzt man den DIM-Befehl (Kürzel für Dimension). Felder müssen vor Gebrauch immer definiert werden! (Dies war nicht der Fall bei den Einwert-Variablen, die wir bislang benutzt haben.)

Die Form für die DIMensionierung numerischer Felder ist:

DIM variablenname (größe)

Dabei bedeutet:

variablenname — Eine Bezeichnung der Variablen gemäß den oben besprochenen Benennungsregeln für numerische Variable.

größe — ist die Anzahl der Speicherplätze, die sich im Bereich zwischen 0 und 255 bewegen muß. Beachten Sie, daß, wenn Sie eine Größenordnung eingeben, ein Speicherplatz mehr bereitgestellt wird.

Beispiele für mögliche DIMensionierungsbefehle sind:

```
DIM X(5)
DIM AA(24)
DIM Q5(0)
```

Der erste Befehl schafft ein Feld X mit 6 Speicherplätzen. Der zweite Befehl baut ein Feld mit 25 Speicherplätzen auf, der dritte ein Feld Q5 mit einem Speicherplatz, was im Grunde unsinnig ist (zumindest für Zahlen), da man ebensogut eine einwertige numerische Variable definieren könnte.

Es ist wichtig zu wissen, daß eine Feld-Variable X und eine Variable X sich unterscheiden und auch vom SHARP PC 1260/1261 unterschieden werden können. Das erste X bezeichnet eine Serie von numerischen Speicherplätzen, das zweite einen einzelnen und getrennten Speicherplatz.

Nachdem Sie nun wissen, wie man Felder aufbaut, mögen Sie sich fragen, wie man die einzelnen Speicherplätze anspricht. Da die gesamte Gruppe unter einem einzigen Namen abgelegt ist, sprechen wir einen einzelnen Speicherplatz ("Element" genannt) an, indem wir an den Namen der Gruppe eine Zahl in Klammern anschließen. Diese Zahl wird "Index" genannt. So müßte man z. B., um die Zahl 9 an fünfter Stelle in unserem (vorher definierten) Feld X unterzubringen, schreiben:

```
X(4) = 8
```

Wenn Sie die Zahl 4 verwirrt, bedenken Sie, daß die Numerierung der Elemente in einem Feld mit Null beginnt und dann bis zu der definierten Obergrenze fortläuft.

Der besondere Vorteil von Feldern liegt in der Möglichkeit, einen längeren Ausdruck oder eine Variable als Index zu benutzen.

Um ein String-Feld zu DIMensionieren, muß man etwas anders vorgehen als bei numerischen Feldern:

DIM stringvariablenname (größe) * länge

Dabei bedeutet:

stringvariablenname — Eine Bezeichnung der Variablen gemäß den oben besprochenen Benennungsregeln für String-Variable.

größe — Anzahl der Speicherplätze (im Bereich zwischen 0 und 255). Beachten Sie, wenn Sie eine Größenordnung angeben, daß ein Speicherplatz mehr bereitgestellt wird.

länge — (fakultativ) Wenn sie angegeben wird, spezifiziert sie die Länge der einzelnen, im Feld enthaltenen Strings. Die Länge muß im Bereich zwischen 1 und 80 liegen. Wird die Länge nicht angegeben, erhalten die Strings automatisch eine Länge von 16 Zeichen.

Beispiele für mögliche DIMensionierungsbefehle für String-Felder sind:

```
DIM X$(4)
DIM NM$(19) *10
DIM IN$(1) * 80
DIM R$(0) *26
```

Im ersten Fall wird ein Feld mit fünf Strings aufgebaut, in denen je 16 Zeichen gespeichert werden können. Der zweite DIM-Befehl baut ein Feld NM auf mit elf Strings zu je 10 Zeichen. Definiert man auch Stringlängen, die kleiner als 16 Zeichen sind, spart man natürlich Speicherkapazität. Das dritte Beispiel zeigt ein Feld IN mit zwei Strings zu je 80 Buchstaben, und im letzten Fall wird ein Feld R mit einem einzigen, 26 Buchstaben umfassenden String aufgebaut.

Neben den einfachen Feldern, die wir gerade behandelt haben, läßt der PC 1260/1261 auch zweidimensionale Felder zu. Ein eindimensionales Feld ist eine Liste von Daten, die in einer einzigen Spalte angeordnet sind. Entsprechend ist ein zweidimensionales Feld eine Tabelle mit Zeilen und Spalten. Zweidimensionale Felder werden folgendermaßen DIMensioniert:

DIM variablenname (zeilen, spalten)

oder

DIM stringvariablenname (zeilen, spalten) * länge

Dabei bedeutet:

zeilen — Gibt die Anzahl der Zeilen im Feld an. Dies muß eine eine Zahl zwischen 0 und 255 sein. Beachten Sie, daß wenn Sie eine Größenordnung eingegeben haben, eine Zeile mehr bereitgestellt wird.

spalten — gibt die Anzahl der Spalten im Feld an. Dies muß eine Zahl zwischen 0 und 255 sein. Beachten Sie, daß, wenn Sie eine Größenordnung eingeben, eine Spalte mehr eingerichtet wird.

Die folgende Tabelle illustriert die Speicherplätze, die sich aus der Anweisung DIM T (2, 3) und den (jetzt aus zwei Zeichen bestehenden) Indizes, die zu den jeweiligen Speicherplätzen gehören, ergibt.

	Spalte 1	Spalte 2	Spalte 3	Spalte 4
Zeile 0	T(0,0)	T(0,1)	T(0,2)	T(0,3)
Zeile 1	T(1,0)	T(1,1)	T(1,2)	T(1,3)
Zeile 2	T(2,0)	T(2,1)	T(2,2)	T(2,3)

Achtung: Zweidimensionale Felder nehmen viel Speicherplatz in Anspruch! Z. B. benötigt ein Feld mit 25 Zeilen und 35 Spalten 875 Speicherplätze.

Felder sind sehr nützliche Programmier-Hilfsmittel.

Die folgende Tabelle zeigt die Anzahl der Bytes, die zur Definierung jeder einzelnen Variable benötigt werden sowie die Anzahl der für jeden einzelnen Programm-Befehl erforderlichen.

Variable	Name der Variablen	Daten	
Numerische Variable	7 Bytes	8 Bytes	
String-Variable	7 Bytes	String-Variable	Spezifizierte Anzahl
		einfache Variable (2-Zeichen-Variable)	16 Bytes

Wenn z. B. DIM Z$(2,3)*10 spezifiziert wird, werden 12 Variable mit einer Speicherkapazität von je 10 Zeichen bereitgestellt. Dies erfordert 7 Bytes (Name der Variablen) + 10 Bytes (Anzahl der Zeichen) x 12 = 127 Bytes.

Element	Zeilen-nummer	Befehl Funktion	Andere, ENTER
Anzahl der verwendeten Bytes	3 Bytes	1 Byte	1 Byte

VARIABLE DER FORM A ( )

Wird im Speicher des Computers ein Bereich für vorgegebene Variable reserviert, kann dieser auch benutzt werden, um mit einem Index versehene Variable (die die gleiche Form wie Feld-Variable haben) zu definieren.

26 vorbestimmte Variablennamen stehen zur Verfügung, nämlich A bis Z (A$ bis Z$). Jeder dieser Namen kann als Index eine der Zahlen von 1 bis 26 tragen, wie z. B. A(1) - A(26) oder A$(1) - A$(26). D. h., daß die Variable A(1) anstelle der Variablen A benutzt werden kann, A(2) anstelle von B, A(3) anstelle von C usw.

Wenn allerdings schon ein Feld mit dem Namen A oder A$ durch einen DIM-Befehl definiert wurde, können Index-Variablen mit den Namen A nicht zusätzlich definiert werden. Wurde beispielsweise ein Feld A definiert durch DIM A(5), sind die Speicherplätze A(0) bis A(5) im Programm-/Datenbereich reserviert. Wenn man dann eine Variable A(2) spezifiziert, bezieht sich diese nicht auf die vorbestimmte Variable B, sondern auf die Feld-Variable A(2) im Programm-/Datenbereich. Spezifizieren Sie A(9), wird dies eine Fehlermeldung bewirken, da A(9) außerhalb des durch den DIM A(5)-Befehl bestimmten Bereiches liegt.

Wenn andererseits Index-Variable der Form A( ) eingegeben worden sind, ist es nicht möglich, gleichzeitig Felder A oder A$ mit Hilfe eines DIM-Befehls einzurichten, es sei denn, die Definition der Index-Variablen wird mit dem CLEAR-Befehl gelöscht.

DER GEBRAUCH VON INDIZES GRÖSSER ALS 26

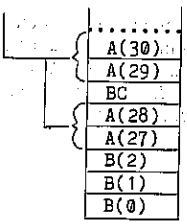
Sollen Indizes größer als 26 für Variable der Form A( ) benutzt werden, wenn ein Feld A nicht durch ein DIM-Statement definiert wurde, werden die entsprechenden Speicherplätze im Programm-/Datenbereich für diese A( )-Variablen definiert. Wenn Sie z. B. A(35) = 5 eingeben, werden die Speicherplätze A(27) bis A(35) im Programm-/Datenbereich reserviert.

Da Variable mit einem Index größer als 26 wie Feld-Variable behandelt werden, unterliegen sie den folgenden besonderen Beschränkungen:

(1) Die Speicherplätze für ein Feld mit demselben Namen müssen im Programm-/Datenbereich nebeneinander liegen. Andernfalls erhalten Sie eine Fehlermeldung.

```
10 DIM B(2)
20 A(28) = 5
30 BC = 12
40 A(30) = 9
```

Wenn dieses Programm eingegeben wird, befinden sich die Definitionen für das Feld "A" nicht in zwei aufeinanderfolgenden Segmenten des Programm-/Datenbereiches und aus Zeile 40 wird sich ein Fehler ergeben.



(2) Numerische Feld-Variable und String-Feld-Variable mit demselben Index können nicht zur gleichen Zeit definiert werden. So können z. B. A(30) und A$(30) nicht zur gleichen Zeit definiert werden, da sie im Programm-/Datenbereich den gleichen Platz einnehmen.

(3) Zweidimensonale Felder können nicht definiert werden, ebensowenig ist es möglich, die Länge der Zeichenstrings zu definieren, die in einer Feld-Variable enthalten sein sollen. Die Länge des Buchstabenstrings, der von der Variablen A$( ) erfaßt werden kann, ist beispielsweise auf 7 oder weniger Zeichen beschränkt.

(4) Variable mit dem Index (0) können nicht definiert werden. Wenn Sie A(0) oder A$(0) definieren, werden Sie eine Fehlermeldung erhalten.

AUSDRÜCKE

Ein Ausdruck ist eine Kombination von Variablen, Konstanten und Operatoren, die durch Rechenoperation auf einen einzigen Wert zusammengefaßt werden kann. Die Rechenbeispiele, die Sie in Kapitel 3 eingegeben haben, waren Beispiele für solche Ausdrücke. Ausdrücke sind ein wesentlicher Bestandteil von BASIC-Programmen. Z. B. kann ein Ausdruck eine Formel sein, mit der das Ergebnis einer Gleichung ermittelt wird, oder ein Test zur Bestimmung des Verhältnisses zwischen zwei Größen, ein Mittel, um eine Reihe von Strings zu formatieren.

NUMERISCHE OPERATIONEN

Der PC 1260/1261 verfügt über fünf numerische Operatoren. Dies sind die arithmetischen Rechenzeichen, die Sie benutzt haben, als Sie die Benutzung des PC 1260/1261 als Taschenrechner in Kapitel 3 kennenlernten.

```
+ Addition
- Subtraktion
* Multiplikation
/ Division
^ Exponentiation
```

Ein numerischer Ausdruck wird in derselben Weise konstruiert, wie Sie komplexe Rechenbefehle eingegeben haben. Numerische Ausdrücke können jede aussagefähige Kombination von numerischen Konstanten, numerischen Variablen und numerischen Operatoren beinhalten:

```
(A*B)^2
A(2,3)4A(3,4)45,0-C
(A/B)*(C+D)
```

STRING-AUSDRÜCKE

String-Ausdrücke sind ähnlich den numerischen Ausdrücken, allerdings gibt es nur einen einzigen String-Operator: die Verkettung (+). Das benutzte Symbol ist dasselbe wie das Plus-Zeichen. Wird es mit einem String-Paar benutzt, knüpft das + den zweiten String an das Ende des ersten an und schafft dadurch einen längeren String. Wenn Sie komplexere String-Verkettungen und andere String-Operationen vornehmen, bedenken Sie bitte, daß der PC 1260/1261 nur 79 Zeichen zur Zeit annimmt.

Achtung: String-Einheiten und numerische Einheiten können nicht in demselben Ausdruck definiert werden, es sei denn, man benutzt eine der Funktionen, die String-Werte in numerische Werte umwandeln oder umgekehrt.

```
"15" + 10      ist unzulässig
"15" + "10"    ist "1510", nicht "25"
```

VERHÄLTNIS-AUSDRÜCKE

Ein Verhältnis-Ausdruck vergleicht zwei Ausdrücke und gibt an, ob das festgestellte Verhältnis wahr oder unwahr ist. Die Verhältnis-Operatoren sind:

```
>    größer als
>=   größer oder gleich
=    gleich
<>   ungleich
<=   kleiner oder gleich
<    kleiner als
```

Die folgenden Ausdrücke werden als Verhältnis-Ausdrücke akzeptiert:

```
A<B
C(1,2)>=5
D(3)<>8
```

Wenn A gleich 10 wäre, B gleich 12, C(1,2) gleich 6 und D(3) gleich 9, wären alle diese Ausdrücke wahr.

Zeichen-Strings können ebenfalls mit Hilfe von Verhältnis-Ausdrücken verglichen werden. Die beiden Strings werden Zeichen für Zeichen gemäß dem Wert ihres ASCII-Codes verglichen (siehe Anhang B: ASCII-Code-Tabelle). Ist ein String kürzer als der andere, wird 0 oder NUL in die freibleibenden Positionen eingesetzt. Alle folgenden Beispiele sind wahr:

```
"ABCDEF" = "ABCDEF"
"ABCDEF" <> "ABCDE"
"ABCDEF" > "ABCDE"
```

Verhältnis-Ausdrücke beurteilen nach wahr oder unwahr. Beim PC 1260/1261 wird "wahr" durch eine 1 angegeben, "unwahr" durch eine 0. In jeder logischen Überprüfung wird ein Ausdruck, der 1 oder mehr ergibt, als wahr betrachtet, während jeder, der 0 ergibt, als unwahr verstanden wird.

LOGISCHE AUSDRÜCKE

Logische Ausdrücke sind Verhältnis-Ausdrücke, die die Operatoren AND, OR und NOT benutzen. AND und OR werden verwendet, um zwei Verhältnis-Ausdrücke miteinander zu verbinden, der Wert des verbundenen Ausdrucks wird aus der folgenden Tabelle ersichtlich:

A AND B

Wert von B \ Wert von A	wahr	unwahr
wahr	wahr	unwahr
unwahr	unwahr	unwahr

A OR B

Wert von B \ Wert von A	wahr	unwahr
wahr	wahr	wahr
unwahr	wahr	unwahr

Dezimalzahlen können folgendermaßen in binärer Schreibweise von 16 Bit dargestellt werden:

Dezimalform	Binärform
32767	0111111111111111
3	0000000000000011
2	0000000000000010
1	0000000000000001
0	0000000000000000
-1	1111111111111111
-2	1111111111111110
-3	1111111111111101
-32767	1000000000000000

Die negative Form einer Dezimalzahl wird folgendermaßen dargestellt:

```
NOT          0000000000000001
(Negativ)    1111111111111110
```

So wird also für jedes Bit 1 zu 0 interpretiert, und 0 zu 1. Dies nennt man "Inversion(NOT)".

Demnach ergibt sich das Folgende, wenn man 1 und NOT 1 addiert:

```
 0000000000000001 (1)
(+) 1111111111111110 (NOT 1)
--------------------
 1111111111111111 (-1)
```

Alle Bits werden also 1. Entsprechend der obigen Zahlenliste werden die Bits in dezimaler Schreibweise -1, d. h. 1 + NOT 1 = -1.

Das Verhältnis zwischen einem numerischen Wert X und seiner negativen Form (NOT X) ist:

$$X + \text{NOT } X = -1$$

Daraus ergibt sich die Gleichung NOT X = -X-1, d. h. NOT X = -(X+1).

Aus dieser Gleichung lassen sich folgende Ergebnisse ableiten:

```
NOT 0 = -1
NOT-1 =  0
NOT-2 =  1
```

Mit diesen logischen Operationen können Sie auch mehr als zwei Ausdrücke in Beziehung setzen. Dann sollten Sie aber darauf achten, Klammern zu verwenden, um den gewünschten Vergleich zu verdeutlichen.

```
(A<9) AND (B>5)
(A>=10) AND NOT (A>20)
(C=5) OR (C=6) OR (C=7)
```

Der PC 1260/1261 behandelt logische Operationen "bitweise" als logische Funktionen auf der Basis von 16 Bit (siehe Anmerkung zu den Verhältnis-Ausdrücken und wahr/unwahr). In normalen Operationen hat dies keine Bedeutung, weil die einfache 1 oder 0 (wahr oder unwahr), die aus Verhältnis-Ausdrücken resultieren, nur je ein Bit einnehmen. Wenn Sie jedoch logische Operatoren auf andere Werte als 0 oder 1 anwenden, wird jedes Bit einzeln bearbeitet. Wenn z. B. A = 17 ist, und B = 22, dann ist (A OR B) = 23.

```
17 ist in binärer Schreibweise 10001
22 ist in binärer Schreibweise 10110
17 OR 22 ist                   10111
10111 in dezimaler Schreibweise ist aber 23.
```

Wenn Sie ein geübter Programmierer sind, kennen Sie sicherlich Anwendungen, wo diese Art der Operation sehr nützlich sein kann. Programmieranfänger sollten sich vorerst an klare, einfache Wahr-/Unwahr-Aussagen halten.

KLAMMERUNG UND VORRANG DER OPERATOREN

Bei der Bearbeitung komplexer Ausdrücke folgt der PC 1260/1261 einer Reihe vordefinierter Prioritäten, die bestimmen, in welcher Reihenfolge die Operatoren bearbeitet werden. Das kann sehr wichtig sein:

5 + 2 * 3 kann sein:

```
5 + 2 = 7          oder        2 * 3 = 6
7 * 3 = 21                     6 + 5 = 11
```

Die genauen Regeln des "Operatoren-Vorrangs" werden in Anhang D beschrieben.

Damit Sie sich nicht alle diese Regeln merken müssen, und damit Sie Ihre Programme eindeutiger gestalten, benutzen Sie immer Klammern, um die Reihenfolge der Bearbeitung von Operatoren vorzugeben. Das obige Beispiel wird eindeutig, wenn Sie schreiben:

(5 + 2) * 3 oder 5 + (2 * 3)

RUN-MODUS

Im allgemeinen kann jeder der oben angesprochenen Ausdrücke ebenso im RUN-Modus benutzt werden wie bei der Programmierung eines BASIC-Statements. Im RUN-Modus wird ein Ausdruck sofort ausgewertet und angezeigt, z. B.

EINGABE	ANZEIGE
(5>3) AND (2<6)	(5>3) AND (2<6) 1.

Die 1 gibt an, daß der Ausdruck wahr ist.

FUNKTIONEN

Funktionen sind spezielle Bestandteile der BASIC-Sprache, die einen Wert in einen anderen Wert transformieren. Funktionen arbeiten wie Variable, deren Wert durch andere Variable oder Ausdrücke festgelegt wurde. ABS ist eine Funktion, die den absoluten Wert der Zahl hervorbringt, auf die sie angewendet wird.

ABS (-5) ist 5
ABS (6) ist 6

LOG ist eine Funktion, die den dekadischen Logarithmus der Zahl berechnet, auf die LOG angewendet wird.

```
LOG (100)  ist 2.
LOG (1000) ist 3.
```

Eine Funktion kann überall dort verwendet werden, wo auch Variable benutzt werden können. Viele Funktionen erfordern keine Klammerung.

```
LOG 100 ist dasselbe wie LOG (100)
```

Sie müssen jedoch Klammern benutzen, sobald sich eine Funktion auf mehr als eine Zahl bezieht. Der Gebrauch von Klammern sorgt immer für eine klarere und eindeutigere Programmierung.

In Kapitel 8 finden Sie eine vollständige Liste der auf dem PC 1260/1261 verfügbaren Funktionen.

KAPITEL 5

PROGRAMMIEREN MIT DEM PC 1260/1261

Im vergangenen Kapitel haben wir einige Begriffe und Ausdrücke der Programmiersprache BASIC kennengelernt. In diesem Kapitel wollen wir nun diese Elemente benutzen, um auf dem PC 1260/1261 Programme zu schreiben. Wir möchten noch einmal darauf hinweisen, daß dieses Handbuch nicht als Einführung in die BASIC-Programmierung verstanden werden soll. Dieses Kapitel soll Sie lediglich in den besonderen Gebrauch des BASIC auf dem PC 1260/1261 einführen.

PROGRAMME

Ein Programm besteht aus einer Reihe von an den Computer gerichteten Befehlen. Denken Sie daran, daß der PC 1260/1261 nur eine Maschine ist, die genau die Operationen durchführt, die Sie angeben. Sie als Programmierer sind dafür verantwortlich, korrekte Befehle einzugeben.

BASIC STATEMENTS

Der PC 1260/1261 setzt Programme entsprechend einem bestimmten Format um. Dieses Format wird Statement genannt. Sie geben die BASIC-Statements immer nach einem bestimmten Muster ein. Ein Statement muß mit einer Zeilennummer beginnen.

```
10: PRINT "HALLO"
```

ZEILENNUMMERN

Jede Programmzeile muß eine eigene Nummer haben, und zwar muß dies eine ganze Zahl zwischen 1 und 65279 sein. Zeilennummern sind die Bezugspunkte des Computers. Sie geben dem PC 1260/1261 an, in welcher Reihenfolge er ein Programm abarbeiten muß. Es ist nicht erforderlich, daß Sie die Programmzeilen folgerichtig eingeben (obwohl dieser Weg sicher der weniger verwirrende ist - besonders, wenn Sie noch Anfänger sind). Der Computer beginnt beim Durcharbeiten eines Programmes immer mit der niedrigsten Zeilennummer und arbeitet die folgenden in aufsteigender Reihenfolge ab.

Beim Programmieren ist es sinnvoll, genug Raum für spätere Einschübe zwischen den einzelnen Zeilen zu lassen (10, 20, 30, ...10, 30, 50 etc.).

Benutzen Sie keine gleichen Zeilennummern in verschiedenen Programmen: Wenn Sie gleiche Zeilennummern benutzen, wird die ältere Zeile gelöscht, sobald Sie eine neue mit derselben Nummer eingeben.
Benutzen Sie nicht die Ziffern 8960 bis 9215 als Zeilennummern. Wenn Sie sie als Zeilennummern benutzen, kann es passieren, daß diese Zahlen nicht korrekt auf der unteren Zeile des Displays wiedergegeben werden. Stattdessen können andere Zeichen angezeigt werden. Wenn dies der Fall ist, benutzen Sie die Pfeil-nach-unten-Taste und verschieben Sie die untere Zeile in die obere Zeile des Displays, um sie korrekt anzuzeigen. Durch die falsch angezeigte Zeile wird die Ausführung oder der Druck des Programms nicht beeinträchtigt.

BASIC-BEFEHLE

Alle BASIC-Statements müssen Befehle enthalten. Diese Befehle sagen dem Computer, welche Operationen er durchführen soll. Ein Befehl ist immer Programmbestandteil, insofern wird die Operation nicht direkt darauf erfolgen.

Einige Statements erfordern oder erlauben den Gebrauch von Operanden:

```
10: PRINT "HALLO"
20: READ B(10)
30: END
```

Operanden informieren den Computer darüber, auf welche Daten sich die vom Befehl angeordnete Operation bezieht. Einige Befehle erfordern Operanden, im Zusammenhang mit anderen Befehlen können (müssen aber nicht) Operanden eingesetzt werden.

(In Kapitel 8 finden Sie eine vollständige Übersicht über die auf PC 1260/1261 möglichen BASIC-Befehle und ihren Gebrauch.)

BASIC-KOMMANDOS

Kommandos sind Anweisungen an den Computer, die außerhalb des Programms eingegeben werden. Kommandos fordern den Computer auf, bestimmte Handlungen mit dem Programm vorzunehmen oder einen bestimmten Modus zu setzen, der dann wiederum die Art der Programmabarbeitung determiniert.

Anders als die Befehle haben Kommandos unmittelbare Wirkung - sobald Sie die Eingabe des Kommandos beendet haben (mit Betätigen der ENTER -Taste), wird das Kommando ausgeführt. Kommandos geht keine Zeilennummer voraus.

```
RUN
NEW
RADIAN
```

Einige Kommandos können auch als Befehle verwendet werden. (In Kapitel 8 findet sich eine vollständige Übersicht der Kommandos und ihres Gebrauchs auf dem PC 1260/1261.)

MODI

Sie erinnern sich sicherlich, daß Sie, als Sie den PC 1260/1261 als Rechner ohne Programmunterstützung benutzt haben, den RUN-Modus gewählt haben.

Der RUN-Modus wird ebenfalls gebraucht, um die von Ihnen geschriebenen Programme abzuarbeiten.

Der PROgramm-Modus wird gewählt, wenn Sie Programme eingeben oder editieren wollen.

Der ReSerVe-Modus ermöglicht es, vordefinierte String-Variable zu kennzeichnen und zu speichern. Weiterhin wird er für komplexere Programmierung gebraucht (siehe Kapitel 6).

DER ANFANG DES PROGRAMMIERENS MIT DEM PC 1260/1261

Nachdem Sie in vielen Übungen den PC 1260/1261 als Taschenrechner benutzt haben, sind Sie sicherlich schon recht vertraut mit der Tastatur. Von nun an werden wir, wenn wir eine Eingabe zeigen, nicht mehr jede Tastenbedienung zeigen. Denken Sie immer daran, [SHIFT] zu betätigen, um Zugang zu den Zeichen oberhalb einzelner Tasten zu bekommen. Vergessen Sie nicht, AM ENDE JEDER ZEILE DIE [ENTER] -TASTE ZU DRÜCKEN!

Nun können Sie mit dem Programmieren beginnen. Bringen Sie den Schiebeschalter in die PROgramm-Stellung und geben das folgende Kommando ein:

EINGABE	ANZEIGE
NEW	>

Das NEW-Kommando löscht aus dem Arbeitsspeicher des PC 1260/1261 alle vorhandenen Daten und Programm. Das Bereitschaftssymbol erscheint, wenn Sie die [ENTER] -Taste gedrückt haben. Es zeigt an, daß der Computer nun Eingaben erwartet.

Beispiel 1:

Vergewissern Sie sich, daß der PC 1260/1261 sich im PRO-Modus befindet, und geben Sie das folgende Programm ein:

EINGABE	ANZEIGE
10 PRINT "HALLO"	10: PRINT "HALLO"

Beachten Sie, daß der PC 1260/1261 ihre Eingabe anzeigt, sobald Sie [ENTER] gedrückt haben, und daß er dabei einen Doppelpunkt (:) zwischen die Zeilennumer und den Befehl einfügt. Vergewissern Sie sich, daß das Statement in der richtigen Form eingegeben wurde.

Nun bringen Sie den Schiebeschalter in die RUN-Stellung:

EINGABE	ANZEIGE
RUN	HALLO

Da dies die einzige Programmzeile ist, wird der Computer an dieser Stelle stoppen. Drücken Sie [ENTER], um aus dem Programm herauszukommen und geben Sie RUN vor, wenn das Programm noch einmal ablaufen soll.

Beispiel 2:
DAS EDITIEREN EINES PROGRAMMS

Nehmen wir an, Sie wollen den Inhalt dessen, was Ihr Programm anzeigt, verändern, d. h. Sie wollen Ihr Programm editieren. Bei einem einzeiligen Programm könnten Sie die Eingabe schlicht wiederholen, aber wenn Sie komplexere Programme schreiben, wird das Editieren zu einem wichtigen Bestandteil des Programmierens. Wir wollen das Programm, das Sie gerade geschrieben haben, editieren.

Sind Sie noch im RUN-Modus? Wenn ja, bringen Sie den Schiebeschalter wieder in die Stellung PRO.

Um Ihr Programm editieren zu können, müssen Sie es nun wieder abrufen. Benutzen Sie dazu die [↑] -Taste. Sofern Ihr Programm vollständig durchlaufen worden war, wird die [↑]-Taste lediglich die letzte Programmzeile abrufen. Falls ein Fehler im Programm vorlag oder Sie die Ausführung mit [BRK] unterbrochen hatten, wird die [↑]-Taste die Zeile abrufen, in der der Fehler liegt oder in der [BRK] betätigt wurde. Um Veränderungen im Programm vorzunehmen, bedienen Sie sich der [↑] -Taste, um sich im Programm nach oben zu bewegen (d. h. die jeweils vorherige Zeile abzurufen), der [↓]-Taste, um sich nach unten zu bewegen (d. h. die jeweils nächste Zeile abzurufen). Wenn die [↑]- und [↓] -Tasten festgehalten werden, "rollt" das Programm im Display nach oben bzw. unten, d. h. die einzelnen Zeilen werden nacheinander durch das Display laufen.

Sie erinnern sich sicherlich, daß Sie den Cursor auf der Zeile mit den [►] - und [◄] -Tasten bewegen. Mit Hilfe des Pfeiles setzen Sie nun den Cursor auf das erste Zeichen, das Sie ändern möchten.

Die Anzeige-Einheit, eine Kombination aus oberer und unterer Zeile, kann 48 Spalten ausgeben. Wenn jedoch 48 oder mehr Spalten in einer Zeile auftauchen, ist der über 48 hinausgehende Teil nicht sichtbar. Wenn Sie diesen betrachten wollen, drücken Sie weiterhin auf die [►] -Taste. Der Cursor bewegt sich in die untere Zeile und der bis dahin unsichtbare Teil erscheint in der unteren Anzeige-Zeile. Mehr als 79 Zeichen können in einer Zeile nicht eingegeben werden.

EINGABE	ANZEIGE
[↑]	10: PRINT "HALLO
[◄][◄][◄][◄][◄]	10: PRINT "HALLO"

Beachten Sie, daß der Cursor die Form eines blinkenden Rechtecks angenommen hat, um so anzuzeigen, daß er auf einem schon bestehenden Zeichen steht.

EINGABE	ANZEIGE
TSCHUESS"!	10: PRINT "TSCHUESS"!

Vergessen Sie nicht, am Ende der Zeile [ENTER] zu drücken. Schalten Sie um in den RUN-Modus.

Dies ist ein neuer Typ von Fehlermeldung. Der Fehler ist nicht nur identifiziert (unser alter Freund, der Syntax-Fehler), sondern auch die Nummer der Zeile, in der er auftritt, wird angegeben.

EINGABE	ANZEIGE
RUN	ERROR 1 IN 10

Schalten Sie zurück in den PROgramm-Modus. Sie müssen unbedingt im PROgramm-Modus sein, um Änderungen im Programm vornehmen zu können. Mit Hilfe des [↑]-Pfeiles rufen Sie nun die letzte Zeile Ihres Programms ab.

EINGABE	ANZEIGE
[↑]	10: PRINT "TSCHUESS"! SYNTAX ERROR

Der blinkende Cursor steht auf der kritischen Stelle. In Kapitel 4 haben Sie gelernt, daß Sie, wenn Sie String-Konstanten eingeben, alle dazugehörigen Zeichen in Anführungszeichen einschließen müssen. Benutzen Sie die [DEL] -Taste, um das Ausrufungszeichen zu löschen.

EINGABE	ANZEIGE
[DEL]	10 PRINT "TSCHUESS"_ SYNTAX ERROR

Nun wollen wir das ! an die richtige Stelle setzen. Wenn man Programme editiert, werden die [DEL]- und [INS]-Funktionen genauso benutzt wie beim Korrigieren einfacher Rechenoperationen ohne Programmunterstützung (siehe Kapitel 3). Mit Hilfe des [◄] -Pfeiles bringen Sie den Cursor auf das erste der Einfügung folgende Zeichen.

EINGABE	ANZEIGE
[◄]	10 PRINT "TSCHUESS"!

Drücken Sie die [INS] -Taste. Ein "□" markiert die Stelle, an der ein neues Zeichen einzugeben ist.

EINGABE	ANZEIGE
INS	10 PRINT "TSCHUESS"

Geben Sie das ! ein. Die Anzeige sieht nun folgendermaßen aus:

EINGABE	ANZEIGE
!	10 PRINT "TSCHUESS!"

Denken Sie daran, ENTER zu drücken, damit die Korrektur in das Programm aufgenommen wird!

Achtung: Wenn Sie eine ganze Zeile aus Ihrem Programm löschen möchten, geben Sie nur die Zeilennummer ein und die ursprüngliche Zeile wird gelöscht.

Beispiel 3:
DER GEBRAUCH VON VARIABLEN BEIM PROGRAMMIEREN

Wenn Ihnen der Gebrauch von numerischen und String-Variablen nicht vertraut ist, lesen Sie noch einmal die entsprechenden Abschnitte in Kapitel 4.

Der Gebrauch von Variablen beim Programmieren erlaubt eine sehr differenzierte Ausnutzung der Fähigkeiten Ihres PC 1260/1261.

Erinnern Sie sich, daß Sie einfache numerische Variable mit Buchstaben von A bis Z belegen.

```
A = 5
```

Um String-Variable zu bestimmen, benutzen Sie ebenfalls einen Buchstaben, gefolgt von einem Dollarzeichen. Benutzen Sie niemals den gleichen Buchstaben, um eine numerische und eine String-Variable zu bestimmen. Sie können nicht A und A$ im selben Programm benutzen.

Erinnern Sie sich, daß einfache String-Variable nicht mehr als 7 Zeichen umfassen können.

```
A$ = "TOTAL"
```

Die einer Variablen zugewiesenen Werte können sich im Laufe eines Programms ändern, sie können jeweils den im Programm eingegebenen oder errechneten Wert annehmen. Eine Möglichkeit der Variablenzuweisung ist, den Befehl INPUT zu benutzen. Im folgenden Programmbeispiel ändert sich der Wert der Variablen A$ aufgrund der Daten, die eingegeben werden, um die Frage "WORT?" zu beantworten. Geben Sie dieses Programm ein:

```
10 INPUT "WORT?";A$
20 B= LEN (A$)
30 PRINT "WORT␣HAT";B;"␣BUCHSTABEN"      ┌─bedeutet: Leerschritt
40 END
```

Da Zeile 30 des Programms länger als 24 Zeichen ist, wird der verbleibende Teil in der folgenden Zeile angezeigt.

Das zweite neue Element in diesem Programm ist der Gebrauch des END-Statements, um die Beendigung des Programms anzuzeigen. END läßt den Computer wissen, daß das Programm beendet ist. Es ist eine gute Übung, beim Programmieren immer das END-Statement zu benutzen.

Wenn Ihre Programme komplexer werden, möchten Sie sie möglicherweise noch einmal durchsehen, bevor Sie sie abarbeiten lassen. Zu diesem Zweck benutzen Sie das LIST-Kommando. LIST, das nur im PRO-Modus benutzt werden kann, bringt das gesamte Programm - angefangen bei der niedrigsten Zeilennummer - zur Anzeige.

Versuchen Sie, dieses Programm auflisten zu lassen.

EINGABE	ANZEIGE
LIST	10: INPUT "WORT?";A$ 20: B= LEN (A$)

Benutzen Sie die[↑]- und[↓]-Pfeile, um sich in Ihrem Programm zu bewegen, bis Sie das gesamte Programm durchgesehen haben. Um eine Zeile noch einmal anzusehen, die mehr als 24 Zeichen enthält, bringen Sie den Cursor an die äußerste rechte Seite des Displays und die weiteren Zeichen werden in der Anzeige erscheinen. Nachdem Sie Ihr Programm kontrolliert haben, lassen Sie es ablaufen:

EINGABE	ANZEIGE
RUN	WORT?_
HILFE	WORT HAT 5. BUCHSTABEN
[ENTER]	>

Dies ist das Ende Ihres Programms. Natürlich können Sie es noch einmal ablaufen lassen, indem Sie RUN eingeben. Auf jeden Fall wäre das Programm unterhaltsamer, wenn es mehr als eine Eingabemöglichkeit enthielte. Wir wollen das Programm nun so modifizieren, daß es weiterläuft, ohne daß man nach jeder Antwort RUN eingeben muß.

Gehen Sie zurück in den PRO-Modus und benutzen Sie die Auf- und Abwärts-Pfeile (oder LIST), um in Zeile 40 zu kommen.

Sie können nun 40 eingeben, um so die gesamte Zeile zu löschen. Sie können aber auch den Cursor auf das E in END setzen. Ändern Sie die Zeile 40 folgendermaßen:

```
40: GOTO 10
```

Nun lassen Sie das modifizierte Programm ablaufen.

Das GOTO-Statement schafft eine Programm-Schleife (d. h. eine bestimmte Operation wird immer wieder durchgeführt). Da Sie der Schleife kein Ende gesetzt haben, wird sie sich nicht unterbrechen. Man spricht hier von einer Endlos-Schleife. Um dieses Programm zu unterbrechen, bedienen Sie die [BRK] -Taste.

Wenn Sie das Programm mit der [BRK] -Taste unterbrochen haben, können Sie es mit dem CONT-Kommando neu starten. CONT steht für CONTinue (fortsetzen). Bei Eingabe des CONT-Kommandos läuft das Programm von da an weiter, wo es mit der [BRK] -Taste unterbrochen wurde.

Beispiel 4:
KOMPLEXERE PROGRAMME

Das folgende Programm berechnet N-Fakultät (N!). Das Programm beginnt mit 1 und berechnet N! bis zu der Grenze, die Sie eingeben. Geben Sie das Programm ein:

```
100 F=1: WAIT 128
110 INPUT "GRENZE? ";L
120 FOR N=1 TO L
130 F=F*N
140 PRINT N,F
150 NEXT N
160 END
```

In diesem Programm sind einige neue Erscheinungen enthalten. Der Befehl WAIT in Zeile 100 kontrolliert den Zeitraum, über den eine Zeile im Display bleibt, ehe das Programm fortgesetzt wird. Die Zahlen und ihre Fakultäten werden angezeigt, wenn sie berechnet werden. Durch das WAIT-Statement bleiben sie etwa 2 Sekunden in der Anzeige, statt darauf zu warten, daß Sie [ENTER] drücken.

Ebenfalls in Zeile 100 tauchen zwei Statements auf, die durch einen Doppelpunkt getrennt werden. Sie können so viele Statements in eine Zeile schreiben, wie Sie möchten, solange Sie die einzelnen Statements durch Doppelpunkte voneinander trennen und die maximale Anzahl von 80 Zeichen pro Zeile (einschließlich [ENTER]) nicht überschreiten. Solche Mehrfach-Statements machen allerdings das Lesen und Korrigieren eines Programms schwieriger, deshalb ist es eine gute Übung, sie nur zu benutzen, wenn die Einzel-Statements sehr kurz und einfach sind oder wenn es einen besonderen Grund dafür gibt, Mehrfach-Statements zu verwenden.

In unserem Programm haben wir in Zeile 120 den Befehl FOR und in Zeile 150 den Befehl NEXT benutzt, um eine Schleife aufzubauen. In Beispiel 3 hatten Sie eine Endlos-Schleife geschaffen, die sich kontinuierlich wiederholte. Mit dieser FOR/NEXT-Schleife addiert der PC 1260/1261 jedes Mal 1 zu N hinzu, wenn der Programmablauf den Befehl NEXT erreicht. Hierbei wird geprüft, ob N die anfangs gesetzte Grenze überschreitet. Ist N kleiner oder gleich der Grenze, beginnt der Programmdurchlauf wieder am Anfang der Schleife. Ist N größer als die Grenze, springt das Programm in Zeile 160 und wird dort beendet.

In einer FOR/NEXT-Schleife können Sie jede numerische Variable benutzen. Sie müssen auch nicht mit 1 anfangen, und Sie können bei jedem Programmschritt jeden beliebigen Betrag dazu addieren. Näheres finden Sie in Kapitel 8.

Wir haben dieses Programm mit Zeilennummern von 100 aufwärts belegt. Eine solche Belegung mit unterschiedlichen Zeilennummern macht es möglich, daß man mehrere Programme zur gleichen Zeit im Arbeitsspeicher hat. Um dieses Programm durchlaufen zu lassen, geben Sie statt RUN 10 ein:

```
RUN 100
```

Außer der Möglichkeit, Programme durchlaufen zu lassen, indem Sie die erste Zeilennummer eingeben, können Sie den Programmen auch einen Buchstaben als Namen geben und sie mit der [DEF]-Taste starten (siehe Kapitel 6).

Sie werden bemerkt haben, daß, solange das Programm abläuft, die BUSY-Anzeige aufleuchtet - auch, wenn sonst nichts in der Anzeige steht. Lassen Sie das Programm noch einige Male durchlaufen und versuchen Sie, L verschiedene Werte zuzuweisen.

DAS SPEICHERN VON PROGRAMMEN

Sie erinnern sich sicherlich, daß Einstellungen, ReSerVe-Funktionen und andere Daten und Funktionen im Speicher Ihres PC 1260/1261 erhalten bleiben, auch wenn Sie ihn abschalten. Programme bleiben ebenfalls erhalten, wenn Sie den Computer abschalten oder er das (durch die AUTO-OFF-Funktion) selbst tut. Selbst, wenn Sie die [BRK] -, [CL] - oder [CA] -Tasten benutzen, werden die Programme nicht gelöscht.

Programme gehen nur aus dem Speicher verloren, wenn Sie einen der folgenden Abläufe ausführen:

- NEW eingeben, ehe Sie mit dem Programmieren beginnen;
- den Computer mit dem ALL-RESET-Schalter initialisieren;
- ein neues Programm mit denselben Zeilennummern schreiben, die Sie bereits an ein im Speicher vorhandenes vergeben haben;
- die Batterien wechseln.

Diese kurze Einführung in das Programmieren mit dem PC 1260/1261 sollte dazu dienen, die aufregenden Möglichkeiten des Programmierens mit Ihrem neuen Computer näher zu beschreiben. Weitere Übungen finden Sie in Kapitel 9.

KAPITEL 6

ABKÜRZUNGEN

Der PC 1260/1261 bietet Ihnen verschiedene Möglichkeiten, die Programmierung zu vereinfachen, indem Sie die Anzahl der Tastenbedienungen und anderer sich wiederholender Abläufe reduzieren. Eine dieser Möglichkeiten besteht darin, Befehle und Kommandos abzukürzen (siehe Kapitel 8).

In diesem Kapitel sollen vor allem zwei Möglichkeiten erörtert werden, mit deren Hilfe Sie unnötige Tastenbedienung vermeiden können: die Benutzung der DEF-Taste und der ReSerVe-Modus.

DIE DEF-TASTE UND DER RESERVE-MODUS

Es wird häufiger vorkommen, daß Sie mehrere verschiedene Programme zur gleichen Zeit im Speicher Ihres PC 1260/1261 abspeichern wollen. (Vergessen Sie nicht, verschiedene Zeilennummern zu vergeben!) Um ein Programm mit dem RUN- oder GOTO-Befehl zu starten, müssen Sie sich normalerweise an die erste Zeilennummer eines jeden Programmes erinnern (siehe Kapitel 8). Aber es gibt eine einfachere Möglichkeit! Sie können jedes Programm mit einem einzelnen Buchstaben benennen und das Programm dann mit nur zwei Tastenbedienungen starten.

Anmerkung: Setzen Sie den Kennbuchstaben, auf den Sie sich später beziehen wollen, in die erste Programmzeile. Der Kennbuchstabe muß in Anführungszeichen eingeschlossen und von einem Doppelpunkt gefolgt sein.

```
10: "A":PRINT "ERSTES"
20: END
80: "B":PRINT "ZWEITES"
90: END
```

Jedes der folgenden Zeichen kann als Kennbuchstabe benutzt werden: A, S, D, F, G, H, J, K, L, =, Z, Y, C, V, B, N, M und die Leertaste (SPC). Wie Sie sehen, sind dies alle Tasten in den unteren beiden Zeilen der Tastatur. Dieser Teil wurde zur Unterscheidung gegenüber dem Rest des Tastenfeldes etwas abgedunkelt.

Anmerkung: Um das Programm ablaufen zu lassen, brauchen Sie nun nicht RUN 80 oder GOTO 10 einzugeben. Sie drücken nur die DEF -Taste und dann den Kennbuchstaben. Im obigen Beispiel würde, wenn Sie DEF und dann "B" drücken würden, "ZWEITES" in der Anzeige erscheinen.

Wenn Sie DEF benutzen, um ein Programm zu starten, werden Variable und Modus-Einstellungen ebenso verarbeitet, als wenn Sie GOTO benutzen würden. In Kapitel 8 finden Sie weitere Einzelheiten.

Um ein anderes Programm mit der DEF -Taste während der Ausführung eines Input-Befehls zu starten, drücken Sie die ENTER - oder BRK -Taste, um den Input-Befehl aufzuheben und das neue Programm zu starten.

RESERVE-MODUS

Ein anderer Weg, mit dem PC 1260/1261 Zeit einzusparen.

Innerhalb des Speichers Ihres PC 1260/1261 sind 48 Zeichen für den "Reserve-Speicher" vorgesehen. Sie können diesen Speicherteil benutzen, um häufig gebrauchte Ausdrücke zu speichern. Um diese wieder abzurufen, brauchen Sie nur zwei Tasten zu bedienen.

Anmerkung: Sie speichern Ausdrücke im ReSerVe-Modus ab, der Wiederabruf zum Gebrauch erfolgt aber im RUN- oder PROgramm-Modus.

* Der MEM-Befehl kann abgespeichert werden, er kann aber nicht ausgeführt werden.

* Der PC 1260/1261 hat einen Reserve-Speicher von 48 Bytes. Sie können also bis zu 48 Bytes, einschließlich der Reserve-Taste, in den Reserve-Speicher eingeben. Ein BASIC-Befehl, eine Funktion, ein Buchstabe oder eine Zahl sind jeweils ein Byte. Beispiel: A: S: + - 1 2 A B SIN COS INPUT RUN ... (je 1 Byte)

* Die Länge eines gespeicherten Strings pro Taste liegt innerhalb 48 Bytes. Es können maximal 80 Tastenbedienungen - einschließlich der Reserve- und der [ENTER] -Taste - abgespeichert werden.

Versuchen Sie, das folgende Beispiel nachzuvollziehen:

Bringen Sie den Schiebeschalter Ihres PC 1260/1261 in die RSV-Stellung. Geben Sie NEW ein und drücken dann die [ENTER] -Taste. Dadurch löschen Sie alle vorher gespeicherten Daten, so, wie Sie mit NEW im PROgramm-Modus alle vorher gespeicherten Programme löschen.

Geben Sie [SHIFT] gefolgt von "A" ein:

EINGABE	ANZEIGE
[SHIFT] A	A:

Beachten Sie, daß das "A" links in der Anzeige erscheint und von einem Doppelpunkt gefolgt wird.

Geben Sie das Wort "PRINT" ein und betätigen Sie die [ENTER] -Taste.

EINGABE	ANZEIGE
PRINT [ENTER]	A: PRINT

Ein Leerschritt erscheint nach dem Doppelpunkt und zeigt an, daß "PRINT" nun im Reservespeicher unter dem Buchstaben "A" gespeichert worden ist.

Schalten Sie den PC 1260/1261 in den PROgramm-Modus um. Geben Sie NEW, gefolgt von [ENTER] ein, um den Programmspeicher zu löschen. Geben Sie 10 als Zeilennummer ein und betätigen Sie dann die [SHIFT] -Taste und die "A"-Taste:

EINGABE ANZEIGE

10 [SHIFT] A | 10 PRINT

Sofort wird das Wort "PRINT" hinter der Zeilennummer in der Anzeige erscheinen.

Im Reserve-Speicher kann jede beliebige Zeichenfolge gespeichert werden. Die gespeicherten Strings können jederzeit im RUN- oder PROgramm-Modus wieder abgerufen werden, indem Sie [SHIFT] drücken und anschließend den Buchstaben, unter dem Sie den String abgespeichert haben. Die dafür zur Verfügung stehenden Tasten sind dieselben wie im Zusammenhang mit DEF erwähnt, also alle Tasten im abgedunkelten Bereich des Tastenfeldes.

Um eine gespeicherte Zeichensequenz zu editieren, schalten Sie den ReSerVe-Modus ein und drücken Sie [SHIFT] , gefolgt von dem Buchstaben, unter dem die Sequenz gespeichert ist. Mit Hilfe der Pfeile (rechts und links), der [DEL] - und der [INS] -Taste können Sie nun genauso editieren wie in den anderen Modi.

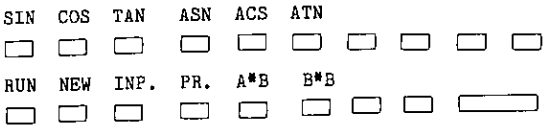
Ist das letzte Zeichen in einer abgespeicherten Sequenz das " @ "-Zeichen, wird es bei Wiederabruf der Sequenz als [ENTER] ausgegeben. Speichern sie beispielsweise den String "GOTO 100" unter der "G"-Taste, so wird bei Betätigung von [SHIFT] und "G" im RUN-Modus augenblicklich die Ausführung des Programms in Zeile 100 beginnen. Ohne das " @ "-Zeichen müssen Sie nach der [SHIFT] "G"-Sequenz [ENTER] eingeben, um das Programm zu starten.

SCHABLONEN

Zu Ihrem PC 1260/1261 gehören zwei Schablonen. Sie können diese benutzen, um das Erinnern an häufig gebrauchte ReSerVe-Sequenzen oder Kennbuchstaben-Zuweisungen über [DEF] zu erleichtern. Nachdem Sie den Programmen Kennbuchstaben zugewiesen haben oder Sequenzen eingerichtet haben, markieren Sie die Schablonen, damit Sie wissen, welche Kennung zu welcher Taste gehört. Sie können dann mit nur zwei Tastenbetätigung Programme abarbeiten lassen oder Sequenzen abrufen.

Wenn Sie beispielsweise eine Reihe von Programmen haben, die Sie häufig zur gleichen Zeit benutzen, vergeben Sie Kennbuchstaben und markieren die Schablonen, so daß Sie die Programmausführung ganz einfach mit nur zwei Tastenbedienungen starten können. Sie können auch häufig gebrauchte BASIC-Befehle und Kommandos in den Reserve-Speicher einspeisen und eine Schablone entsprechend markieren, um so die Eingabe von BASIC-Programmen zu beschleunigen.

Beispiel:



KAPITEL 7

DER GEBRAUCH DES DRUCKERS/MIKROCASSETTENRECORDERS CE-125 UND ANDERER OPTIONEN

Die folgenden Zusatzgeräte können an den PC 1260/1261 angeschlossen werden:

CE-125	Drucker/Mikrocassettenrecorder
CE-126P	Drucker/Cassetten-Interface
CE-152	Cassettenrecorder (CE-126P oder CE-124 erforderlich)
CE-124	Cassetten-Interface (CE-152 kann benutzt werden)

DER GEBRAUCH DER OPTION CE-125 (DRUCKER/MIKROCASSETTENRECORDER)

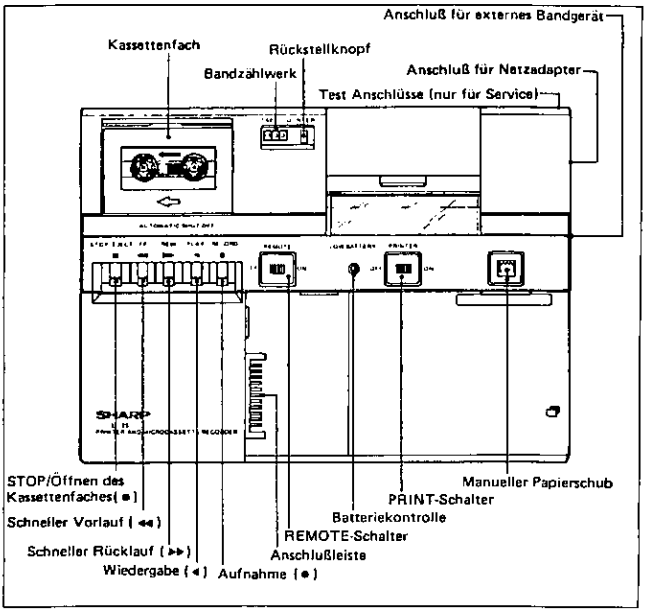
Der CE-125 Drucker/Mikrocassettenrecorder hat ein integriertes Interface für externe Cassettenrecorder. Die Charakteristika der Option CE-125:

* Thermodrucker mit einer Schreibbreite von 24 Zeichen pro Zeile und einer Geschwindigkeit von ca. 48 Zeilen pro Minute
* Handliche Papierzuführung und Abreißvorrichtung
* Auf Wunsch simultaner Ausdruck von Berechnungen
* Einfache Kontrolle der Anzeige oder der Druckerausgabe in BASIC
* Kompletter eingebauter Mikrocassetten-Playback-Recorder mit Schnellvorlauf und Bandzählwerk
* Manuelle und programmgesteuerte Kontrolle des Recorders zum Abspeichern von Programmen, Daten und Reservetastenbelegungen
* Zur besseren Kontrolle und Sicherheit können Dateinamen und Passwörter auf Band gespeichert werden.
* Anschluß für externes Wiedergabegerät
* Eingebaute, wiederaufladbare Nickel-Cadmium-Batterien für tatsächliche Netzunabhängigkeit
* Netz- und Ladeteil (EA-23E) lieferbar
* Automatische Abschaltung des Bandlaufes, wenn das Ende des Bandes erreicht ist (nur im PLAY- und RECORD-Modus).

EINFÜHRUNG IN DAS GERÄT

Bevor Sie beginnen, mit der Option CE-125 zu arbeiten, sollten Sie sich zunächst mit ihren Bestandteilen vertraut machen. Wir wollen zuerst die Vorderseite des Gerätes betrachten:

Die Druckerbefehle sind nur auf der Option CE-125 Drucker/Mikrocassettenrecorder anwendbar. Wird ein Druckbefehl gegeben, wenn die Option CE-125 abgeschaltet ist (Schalter in OFF-Position), erhalten Sie eine Fehlermeldung (ERROR 8). In diesem Fall müssen Sie den Schalter in die Position ON bringen und die [CL] -Taste betätigen. Geben Sie dann erneut den Druckbefehl.



In der rechten unteren Ecke des Gerätes befindet sich die Vertiefung, in der Sie den PC 1260/1261 an die Option CE-125 anschließen. Der Stecker an der linken Seite dieses Bereiches paßt in die entsprechende Buchse am PC 1260/1261. Beachten Sie die Bedienungselemente im dunklen Teil des Geräts. Von links nach rechts sind dies:

* Bedienungselemente des Mikrocassettenrecorders: STOP/EJECT, FF, REW, PLAY und RECORD. Diese dürften Ihnen von gängigen Cassettenrecordern vertraut sein.

* REMOTE-Schalter: Dieser Schalter wird benutzt, wenn Sie den Mikrocassettenrecorder manuell bedienen wollen.

* LOW BATTERY Anzeige: zeigt an, wenn die Stromversorgung nicht ausreicht, um den CE-125 Recorder zu betreiben.

* PRINTER ON/OFF: Dieser Schalter wird benutzt, um den Drucker an- bzw. auszuschalten und so die Batterien zu schonen, wenn der Drucker nicht in Betrieb ist.

* Manueller Papierschub: Wenn Sie diesen Schalter betätigen, wird dem Drucker Papier zugeführt.

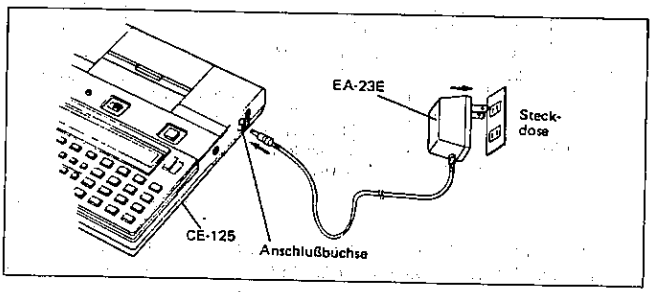
Am oberen Ende des Gerätes, zwischen dem Cassettenlaufwerk und dem Drucker, befindet sich das Bandzählwerk.

STROMVERSORGUNG

Die Option CE-125 wird über eine wiederaufladbare Ni-Cad-Batterie gespeist. Diese muß unbedingt aufgeladen werden, wenn die LOW BATTERY Anzeige aufleuchtet.

Um die Batterie aufzuladen, schalten Sie den Computer und den Drucker/Recorder aus, verbinden Sie das AC-Netzteil (EA-23E) mit dem Drucker/Recorder und stecken Sie das Netzteil in eine Steckdose. Es dauert etwa 15 Stunden, bis die Batterie ganz gefüllt ist.

Wichtige Anmerkung: Die Benutzung eines anderen als des dazugehörigen Netzteils kann den Drucker/Recorder beschädigen!



Stecken Sie das Netzteil immer zuerst in die Option CE-125 ein und erst danach in die Steckdose!

Wenn die Batterien des CE-125 aufgeladen werden müssen, leuchtet die LOW BATTERY Anzeige auf der Vorderseite des Gerätes auf und der Drucker/Recorder ist nicht betriebsbereit. Es ist möglich, daß durch die Lagerung des Gerätes bei der ersten Inbetriebnahme die Batterien nicht genug Spannung liefern. Dann müssen Sie die Einheit vor dem ersten Gebrauch aufladen.

Anmerkung: Wenn Sie den Computer an die Option CE-125 angeschlossen haben und die Batterien des PC 1260/1261 schwach werden, wird dieser von der Option CE-125 mitgespeist.

DAS ANSCHLIESSEN DES PC 1260/1261 AN DIE OPTION CE-125

Um den PC 1260/1261 Taschencomputer an den CE-125 Drucker/Mikrocassettenrecorder anzuschließen, gehen Sie bitte folgendermaßen vor:

1. Schalten Sie die Stromversorgung beider Geräte ab.

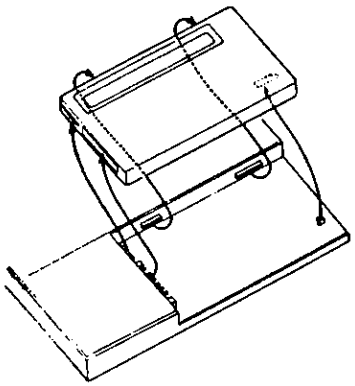
Achtung: Wenn Sie die Stromversorgung nicht abschalten, kann sich Ihr Computer möglicherweise "aufhängen". Wenn dies geschieht, müssen Sie den ALL-RESET-Schalter betätigen, um den Computer neu zu initialisieren.

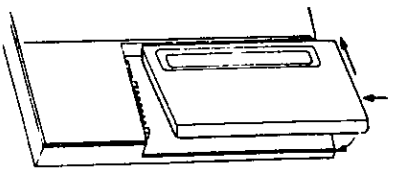
2. Setzen Sie die Oberkante des Computers so in die Vertiefung des Druckers/Recorders, daß die Führungsschienen in die Ausnehmungen am Computer passen.

3. Legen Sie den Computer flach in die Vertiefung.

4. Schieben Sie den Computer vorsichtig nach links, so daß sich der Stecker des Druckers/Recorders in die Buchse des Computers schiebt.

   Drücken Sie den Computer NICHT MIT GEWALT in die Vertiefung am Drucker/Recorder. Wenn die Teile nicht leicht ineinander gleiten, überprüfen Sie, ob alle Teile richtig zusammengefügt sind.



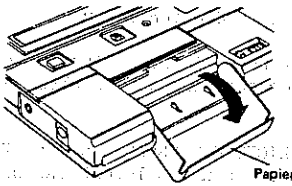


5. Um den Drucker zu benutzen, schalten Sie zuerst den PC 1260/1261 ein, dann den CE-125. Drücken Sie die [CL]-Taste. Wenn die [CL]-Taste nicht betätigt wird, arbeitet der Drucker möglicherweise nicht.

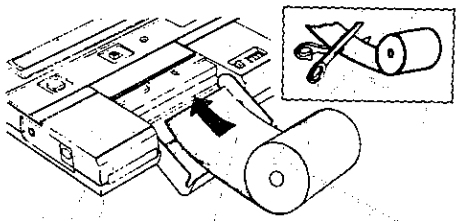
Achtung: Wird ein Druckbefehl gegeben, solange der Drucker abgeschaltet ist, erhalten Sie eine Fehlermeldung (ERROR 8). In diesem Moment kann auch die LOW BATTERY Anzeige aufleuchten. In diesem Fall müssen Sie den Drucker einschalten, die [CL]-Taste drücken und den Druckbefehl noch einmal geben.

DAS EINLEGEN DES PAPIERS

(1) Schalten Sie die Stromversorgung ab.
(2) Öffnen Sie die Papierabdeckung.

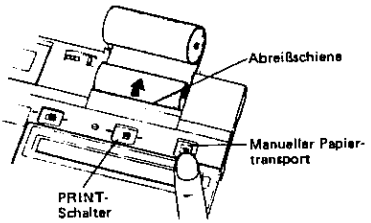


(3) Schieben Sie das lose Ende der Papierrolle in den Schlitz der Papierführung. (Ist das Papier verknickt oder eingerissen, kann dies das Einlegen erschweren!)



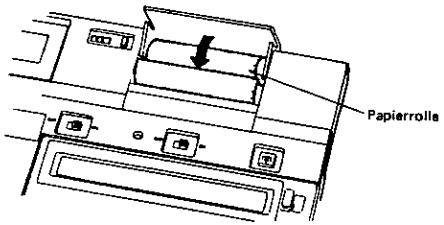
Achtung: Die Benutzung unregelmäßiger Papierrollen kann zu unregelmäßiger Papierzuführung oder Papierstau führen. Sorgen Sie daher unbedingt für gleichmäßig stramm aufgewickelte Papierrollen!

(4) Schalten Sie den Drucker ein und drücken Sie die Manuelle-Papierschub-Taste, bis das Papier aus dem Druckermechanismus herausgeführt wird.



(5) Legen Sie die Papierrolle in die dafür vorgesehene Papierwanne ein.

(6) Schließen Sie die Papierabdeckung.



* Wenn es nötig wird, die Papierrolle zu entfernen, schneiden Sie das Papier auf der Seite der Papierwanne ab und ziehen Sie das verbleibende Papier in der normalen Schubrichtung aus dem Drucker heraus. Ziehen Sie das Papier nicht rückwärts, weil dies den Druckermechanismus beschädigen kann.

Achtung: Papierrollen erhalten Sie dort, wo Sie auch den Drucker CE-125 bezogen haben. Bitte verlangen Sie die Produktnummer EA-1250P (5 Rollen im Paket), wenn Sie Papierrollen nachbestellen. Dieses Papier ist speziell für den CE-125 Drucker ausgelegt. Der Gebrauch von anderem Papier kann das Gerät beschädigen.

DER GEBRAUCH DES DRUCKERS

Wenn Sie den PC 1260/1261 wie einen Taschenrechner benutzen, können Sie mit dem CE-125 Drucker Ihre Berechnungen gleichzeitig ausdrucken. Dies erreichen Sie, indem Sie einfach die [SHIFT] -Taste und dann die [ENTER] -Taste (P↔NP) betätigen, während sich der Computer im RUN-Modus befindet. Die "PRINT"-Anzeige leuchtet auf. Wenn Sie nun am Ende einer Berechnung die [ENTER] -Taste drücken, wird der Inhalt der Anzeige in einer Zeile vom Drucker ausgegeben, das Ergebnis erscheint in der nächsten Zeile des Ausdrucks z. B.:

EINGABE	AUSDRUCK
300/50 [ENTER]	300/50 6

Sie können auch mit Hilfe des LPRINT-Befehls Daten aus einem BASIC-Programm ausdrucken lassen (Einzelheiten siehe Kapitel 8). Der LPRINT-Befehl funktioniert genau wie der PRINT-Befehl, da sowohl die Anzeige als auch die Schreibbreite des Druckers 24 Zeichen umfassen. Der einzige Unterschied besteht darin, daß Sie, wenn Sie mit PRINT arbeiten, alles, was über die 24 Zeichen hinausgeht, nicht sehen können. Arbeiten Sie jedoch mit LPRINT, werden die übrigen Zeichen in einer zweiten und - falls erforderlich - in einer dritten Zeile ausgegeben.

Programme, die mit PRINT geschrieben wurden, können druckfähig gemacht werden, indem Sie ein PRINT=LPRINT-Statement ins Programm schreiben (siehe Kapitel 8). Alle diesem Statement folgenden PRINT-Befehle werden nun behandelt, als wären sie LPRINT-Befehle. Mit dem PRINT=PRINT-Statement versetzen Sie den Computer wieder in den ursprünglichen Zustand. Diese Struktur kann auch mit einem IF-Befehl in ein Programm eingeschlossen werden, wodurch Sie sich zum Zeitpunkt der Programmdurchführung eine Auswahlmöglichkeit schaffen (siehe das Beispiel zum Verhältnis zweier Variablen in Kapitel 9).

Sie können Ihr Programm mit dem LLIST-Kommando vom Drucker auflisten lassen (Einzelheiten siehe Kapitel 8). Wenn Sie LLIST ohne Zeilennummern verwenden, werden alle in dem Moment im Speicher befindlichen Programme nach der numerischen Reihenfolge ihrer Zeilennummern ausgegeben. Die auszugebenden Zeilen lassen sich begrenzen, wenn Sie die entsprechende Spanne der Zeilennummern mit dem LLIST-Kommando eingeben. Wenn Programmzeilen länger als 24 Zeichen sind, werden für den Ausdruck einer Programmzeile zwei, drei oder mehr Zeilen verwendet. Die zweite und alle folgenden Zeilen werden um vier Zeichen eingerückt, damit die Zeilennummer klar zu identifizieren ist.

Achtung:

* Falls eine Fehlermeldung (ERROR 8) infolge eines Papierstaus auftritt, reißen Sie das Papier ab und ziehen Sie das verbliebene Papier vollständig aus dem Drucker. Drücken Sie dann die [CL] -Taste, um die Fehlermeldung zu löschen.

* Wenn der Drucker/Recorder starken externen elektrischen Störfeldern ausgesetzt ist, können eventuell falsche Zahlen willkürlich ausgedruckt werden. In diesem Fall drücken Sie die [BRK] -Taste, um den Druckvorgang zu unterbrechen, danach betätigen Sie die [CL] -Taste. Durch Druck auf die [CL] -Taste setzen Sie den Drucker in den ursprünglichen Zustand zurück.

  Wenn der Drucker einen Papierstau produziert , kann er möglicherweise nicht ordnungsgemäß arbeiten und in der Anzeige erscheint lediglich die BUSY-Meldung. In diesem Fall betätigen Sie die [BRK] -Taste, beseitigen gegebenenfalls den Papierstau und drücken dann die [CL] -Taste.

* Wenn die Option CE-125 nicht gebraucht wird, schalten Sie das Gerät bitte ab, um die Lebensdauer der Batterien zu schonen.

* Auch wenn Sie mit dem LPRINT-Befehl ausdrucken lassen, können Sie Eingaben machen, sofern gerade ein INPUT- oder INKEY$-Befehl ausgeführt wird.

  In diesem Fall wird jedoch der Drucker bei Druck auf die [CL] -Taste unterbrochen. Achten Sie darum darauf, erst bei Ende des Druckvorganges die [CL] -Taste zu drücken.

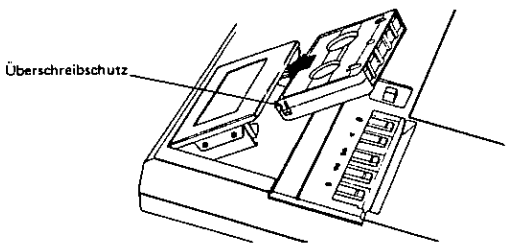
DER GEBRAUCH DES MIKROCASSETTENRECORDERS

DAS EINLEGEN DER TONBANDCASSETTE

1. Drücken Sie die STOP/EJECT-Taste, um den Laufwerksdeckel zu öffnen.

2. Schieben Sie die Cassette in das Laufwerk. Dabei soll die zu benutzende Seite der Cassette ("A" oder "B") nach oben zeigen. Die offene Seite der Cassette zeigt nach vorn.

3. Drücken Sie den Deckel des Laufwerks wieder hinunter.



DAS ENTFERNEN DER TONBANDCASSETTE

1. Drücken Sie die STOP/EJECT-Taste, um den Deckel des Laufwerks zu öffnen. Entfernen Sie dann die Cassette.

2. Drücken Sie den Deckel wieder herunter.

Achtung: Im PLAY-Modus drücken Sie die STOP/EJECT-Taste einmal, um den Lauf der Cassette zu unterbrechen. Erst auf einen zweiten Druck öffnet sich der Laufwerksdeckel.

Benutzen Sie die Tasten zur manuellen Bedienung, um das Band einzulegen. Bringen Sie dann den "REMOTE"-Schalter in die OFF-Position. In dieser Position können Sie die FF-(Schneller Vorlauf) und REW-(Rücklauf)Tasten in Kombination mit dem Bandzählwerk benutzen. Um die Kontrolle der Cassetteneinheit dem Computer zu übertragen, bringen Sie den REMOTE-Schalter in die ON-Position.

Folgende Möglichkeiten haben Sie mit Ihrem Cassettenrecorder:

CSAVE	speichert den Inhalt von Programmen oder Reservespeicher auf Band.
CLOAD	ruft Programme oder Reservespeicherinhalte von Band ab.
CLOAD?	vergleicht das auf Band gespeicherte Programm mit einem im Speicherinhalt, um festzustellen, ob die Kopie korrekt und vollständig ist.
MERGE	kombiniert ein auf Band gespeichertes Programm mit einem im Speicher des Computer befindlichen.
PRINT #	speichert den Inhalt von Variablen auf Band.
INPUT #	ruft den Inhalt von Variablen von Band ab.
CHAIN	startet die Durchführung eines Programms, das auf Band gespeichert ist.

Programmen können Dateinamen zugewiesen werden, die Sie ebenfalls auf Band abspeichern können. Dies schützt Sie vor Verwechslungen, wenn Sie mehrere Programme auf demselben Band abgespeichert haben. Die Programme können dann über ihren Namen abgerufen werden, wobei der Computer auf dem Band nach dem richtigen Programm sucht. Wenn Programme im Speicher mit Paßwörtern geschützt sind, können sie nicht auf Cassette gespeichert werden. Sie können aber Paßwörter in dem Moment zuweisen, in dem Sie Programme mit CSAVE auf Cassette abspeichern. Auf diese Weise gespeicherte Programme können zwar von anderen Personen benutzt, aber nicht gelistet oder verändert werden.

In Kapitel 8 finden Sie Einzelheiten zu den hier erwähnten Befehlen und Kommandos.

Wenn Sie Programme oder Daten auf Band aufnehmen, wird der Aufnahme eine hoher Ton von ca. 7 Sekunden Länge vorausgehen. Dieser Ton dient dazu, das Band über einen "Vorspann" zu befördern und den Anfang des jeweiligen Programms bzw. Datensatzes zu identifizieren.

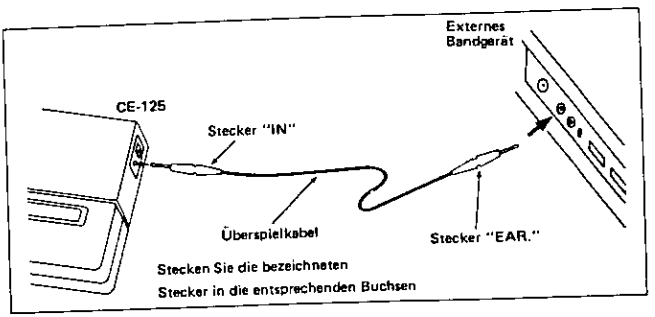
Achtung: Wenn Sie etwas vom Band einlesen wollen, muß das Band auf dem von diesem Ton eingenommenen Bereich stehen!

Wenn nach einem Dateinamen gesucht wird, muß das Band unbedingt vorwärts laufen. Die Suche geht relativ langsam vor sich, von daher ist es manchmal nützlich, sich die Position bestimmter Programme mit Hilfe des Bandzählwerks zu merken. Das Band kann dann mit dem schnellen Vorlauf, dem Rücklauf und der PLAY-Funktion an die richtige Stelle gebracht werden. Während das Band abgesucht wird, können Sie die jeweiligen hohen Kennungstöne hören. Zwischen diesen hohen Tönen zeigen unterschiedliche hohe und tiefe Töne Programme und Daten an.

DER GEBRAUCH EINES EXTERNEN WIEDERGABEGERÄTES

An der Option CE-125 ist auch ein Anschluß für ein externes Wiedergabegerät vorgesehen. Ein solches externes Gerät kann allerdings nur benutzt werden, um Daten vom Band einzulesen, d. h. die CLOAD-, MERGE- und INPUT#-Befehle zu benutzen. Der Hauptgrund für diesen Anschluß liegt in der Möglichkeit, Programme laden zu können, die auf einem anderen SHARP Taschencomputer geschrieben wurden.

Benutzen Sie das zum Gerät gehörige Kabel, um den CE-125 an eine externe Wiedergabeeinheit anzuschließen.



Wird eine externe Wiedergabeinheit angeschlossen, nimmt sie automatisch die Stelle des eingebauten Mikrocassettenrecorders ein und kann mit den entsprechenden Befehlen in der gleichen Weise benutzt werden.

Um Programme und Daten vom Band der externen Wiedergabeeinheit zu überspielen, benutzen Sie bitte das Bandgerät, mit dem die entsprechenden Daten und Programme aufgezeichnet wurden. Andere Geräte arbeiten möglicherweise nicht.

PFLEGE UND WARTUNG

* Gehen Sie sicher, daß die Stromversorung an beiden Geräten abgeschaltet ist, wenn Sie den CE-125 an den PC 1260/1261 anschließen oder abnehmen.

* Der Drucker sollte auf einer Unterlage stehen.

* Die Option CE-125 sollte von extremen Temperaturschwankungen, Feuchtigkeit, Staub und starkem Lärm ferngehalten werden.

* Lassen Sie keine Fremdkörper in den CE-125 gelangen.

* Reinigen Sie die Tonköpfe regelmäßig mit handelsüblichem Reinigungsmittel.

FEHLER

Wenn die Batterien schwach werden oder der CE-125 starkem Lärm ausgesetzt ist, kann die Einheit außer Betrieb geraten und der PC 1260/1261 kann sich "aufhängen". Dies kann auch geschehen, wenn die Einheiten verbunden werden und dabei die Stromversorgung des CE-125 nicht abgeschaltet wurde oder wenn gerade ein LPRINT- oder LLIST-Befehl gebraucht wurde. In einigen Fällen erscheint dann zusätzlich die Meldung ERROR 8 in der Anzeige.

Normalerweise können Sie diese Fehlfunktionen mit der CLear-Taste beheben, unter Umständen kann aber auch das Betätigen des ALL-RESET-Schalters nötig werden. Vergewissern Sie sich, daß Sie die Stromversorgung des CE-125 wieder einschalten, ehe Sie ihn wieder in Betrieb nehmen.

BEISPIELE

DIE ABLÄUFE FÜR COMPUTER- UND MIKROCASSETTENRECORDER-BEDIENUNG

1. Abspeichern auf Cassette

(1) Schalten Sie den Remote-Schalter auf OFF.
(2) Legen Sie ein Band in das Laufwerk ein. (Evtl. das Band manuell straffen!)
(3) Schalten Sie den REMOTE-Schalter auf ON.
(4) Drücken Sie die RECORD-Taste.
(5) Zusammen mit dem Befehl, mit dem Sie Ihr Programm abspeichern, müssen Sie auch einen "Dateinamen" vergeben, damit Sie später einen Bezugspunkt haben. Ihr Dateiname darf nicht länger als 7 Zeichen sein. Um ein Programm mit einem Dateinamen abzuspeichern, geben Sie ein:

CSAVE [SHIFT] [2] PRO-1 [SHIFT] [2]

Ihr Programm wird nun unter dem Namen "PRO-1" abgespeichert. Sie können jeden gewünschten Namen zuweisen, je nachdem, was für Sie gedächtnistechnisch am günstigsten ist. Aber bedenken Sie bitte, daß die Obergrenze für Dateinamen bei 7 Zeichen liegt. Ist Ihr Dateiname länger, wird alles, was über die 7 Zeichen hinausgeht, ignoriert. Es ist sinnvoll, wenn Sie sich ein Programmverzeichnis anlegen, in dem Programmname, Anfang und Ende des Programms auf Band (nach Zählwerk) und eine kurze Beschreibung der Funktionsweise des Programms verzeichnet werden.

Drücken Sie die [ENTER] -Taste. Sie sollten jetzt einen schrillen Summton hören und das Band sollte zu laufen anfangen. Außerdem leuchtet die BUSY-Anzeige auf. Dadurch werden Sie darauf hingewiesen, daß Ihr Computer damit beschäftigt ist, Ihr Programm aus dem Speicher auf das Band zu übertragen. Wenn dies alles nicht geschieht, beginnen Sie noch einmal von vorn.

Sobald der Computer am Ende des Programms angekommen ist, erlischt die BUSY-Anzeige, das Band hört zu laufen auf und das Bereitschaftssymbol erscheint in der Anzeige. Um sich zu vergewissern, daß dies tatsächlich geschehen ist, können Sie den Bandinhalt wieder zurück in den Speicher einlesen lassen. Dies wird im nächsten Abschnitt erklärt.

Anmerkung: Wenn Sie ein Programm auf einer bereits benutzten Cassette abspeichern wollen, löschen Sie ein etwa 5 Zählwerksnummern umfassendes Stück, ehe Sie zu schreiben anfangen. Verfahren Sie dann wie oben beschrieben.

(Vergewissern Sie sich, daß das vorhergehende Programm vollständig gelöscht ist.!)

2. Vergleichen von Speicher- und Bandinhalten

Nachdem Ihr Programm nun auf Band gespeichert worden ist, wollen Sie zweifellos überprüfen, ob es tatsächlich dort ist. Dies können Sie

relativ einfach mit Hilfe des CLOAD?-Befehls tun.

(1) Bringen Sie den REMOTE-Schalter in die Position OFF, um die Kontrolle durch den Computer abzuschalten.

(2) Spulen Sie das Band bis zu dem Punkt zurück, an dem Sie mit dem Abspeichern begonnen haben. Dies ist am einfachsten über das Zählwerk zu leisten.

(3) Schalten Sie die REMOTE-Funktion wieder ein.

(4) Drücken Sie die PLAY-Taste.

(5) Um das Programm mit einem vorab gespeicherten Programm mit Dateinamen zu vergleichen, geben Sie ein:

CLOAD [SHIFT] [?] [SHIFT] ["] PRO-1 [SHIFT] ["]

Drücken Sie nun die [ENTER] -Taste.

Während des Vergleichsvorganges wird das Zeichen "*" in der äußersten rechten Stelle in der unteren Zeile der Anzeige ausgegeben. Die Anzeige verschwindet, wenn der Vergleich abgeschlossen ist. Solange nach einem Dateinamen gesucht wird, steht das "*"-Zeichen nicht in der Anzeige, da der Vergleichsvorgang noch nicht begonnen hat. Das gleiche gilt für die Befehle MERGE, CHAIN, CLOAD und INPUT= sowie in dem Fall, daß das erte Programm ohne einen Dateinamen gelesen wird.

Anmerkung: Wird der vorgegebene Dateiname nicht gefunden, wird der Computer die Suche fortsetzen, auch wenn das Band nicht mehr läuft. In diesem Fall brechen Sie die Suche mit der [BRK] -Taste ab. Das gleiche gilt für die Befehle MERGE, CHAIN, CLOAD und INPUT=.

Der Computer vergleicht das mit CSAVE abgespeicherte Programm mit dem in seinem Speicher befindlichen. Wenn beide übereinstimmen, erscheint das Bereitschaftssymbol in der Anzeige und der Vergleich ist damit beendet. Wurden Fehler in der Übereinstimmung entdeckt, erhalten Sie eine Fehlermeldung, in der Regel ERROR 8. Dies sagt Ihnen, daß das Programm auf der Cassette in irgendeinem Punkt von dem im Speicher befindlichen abweicht. Löschen Sie dann den entsprechenden Teil auf der Cassette und beginnen Sie von vorn.

3. Einladen von Band

(1) Schalten Sie den REMOTE-Schalter auf OFF.
(2) Spulen Sie das Band an Ihren Ausgangspunkt zurück, wobei Sie das Bandzählwerk zu Hilfe nehmen sollten.
(3) Halten Sie den Rücklauf an.
(4) Schalten Sie den REMOTE-Schalter wieder auf ON.
(5) Drücken Sie die PLAY-Taste.

(6) Geben Sie ein:

CLOAD [SHIFT] ["] PRO-1 [SHIFT] ["]

Und drücken Sie die [ENTER] -Taste.

(Sie erinnern sich: "PRO-1" ist der Dateiname, den Sie an Ihr Programm vergeben hatten. Haben Sie Ihr Programm unter einer anderen Kennung abgespeichert, müssen Sie diese anstelle von "PRO-1" eingeben.)

(7) Das Zeichen "*" erscheint in der Anzeige, während das abgespeicherte Programm in den Speicher des Computers eingeladen wird.
(Das gleiche geschieht, wenn das erste Programm ohne Namen gelesen wird. Das Zeichen "*" verschwindet, sobald der Ladevorgang abgeschlossen ist.)

(8) Das eingeladene Programm verbleibt weiterhin auf der Cassette, so daß Sie den Vorgang mit CLOAD beliebig oft wiederholen können.
Wenn während des Ladevorganges die Meldung "ERROR 8" in der Anzeige erscheint, müssen Sie noch einmal bei Schritt (1) anfangen.

DER GEBRAUCH DES CE-126P DRUCKER/CASSETTENRECORDER-INTERFACE

Das Zusatzgerät CE-126P Drucker/Cassettenrecorder-Interface erlaubt den Anschluß eines Druckers oder Cassettenrecorders an Ihren PC 1260/1261.

Die Charakteristika des CE-126P sind:

* Thermodrucker mit 24 Zeichen Schreibbreite
* Handliche Papierzuführungs- und Abreißvorrichtung
* Gleichzeitiges Ausdrucken von Rechnungen, wenn gewünscht
* Einfache Kontrolle der Anzeige oder Druckerausgabe in BASIC
* Eingebautes Cassetten-Interface mit REMOTE-Funktion
* Manuelle oder programmgesteuerte Kontrolle des Recorders zum Abspeichern von Programmen und Daten
* Stromversorgung über Trockenbatterien für tatsächliche Tragbarkeit

Um den PC 1260/1261 an das CE-126P anzuschließen, bedienen Sie sich bitte des mit dem CE-126P gelieferten Handbuches.

DER GEBRAUCH DES DRUCKERS

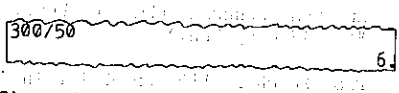
Wenn Sie mit dem PC 1260/1261 Rechnungen ohne Programmunterstützung durchführen, können Sie diese über den CE-126P gleichzeitig ausdrucken lassen.

Sie erreichen dies durch einfaches Drücken der [SHIFT] -Taste und anschließend der [ENTER] -Taste, wobei sich der Computer im RUN-Modus befinden muß.

Die Druckeranzeige (ein waagerechter Strich) erscheint dann direkt über dem Schriftzug "PRINT" im rechten unteren Bereich des Anzeigefeldes. Wenn Sie nun am Ende einer Berechnung die [ENTER] -Taste drücken, wird der Inhalt der Anzeige vom Drucker ausgegeben. Dabei steht die Berechnung selbst in der ersten Zeile, das Ergebnis erscheint in der zweiten. Zum Beispiel:

EINGABE AUSDRUCK

300/50 [ENTER]



Mit dem LPRINT-Befehl können Sie auch Teile von BASIC-Programmen ausdrucken lassen (Einzelheiten dazu finden Sie in Kapitel 8). LPRINT kann in derselben Weise wie der PRINT-Befehl benutzt werden. Der Unterschied besteht darin, daß Sie, wenn Sie einen Ausdruck von mehr als 24 Zeichen mit PRINT drucken wollen, die über 26 hinausgehenden Zeichen nicht sehen können. Mit dem LPRINT-Befehl werden diese Zeichen in einer zweiten und, wenn erforderlich, einer dritten Zeile ausgegeben.

Programme, die mit einem PRINT-Befehl geschrieben wurden, können modifiziert werden, indem Sie in das Programm ein PRINT=LPRINT-Statement hineinschreiben (Einzelheiten siehe Kapitel 8). Alle diesem Statement folgenden PRINT-Befehle werden nun wie LPRINT-Befehle gewertet. Durch PRINT=PRINT heben Sie das Statement wieder auf. Diese Struktur kann auch mit einem IF-Statement in ein Programm geschrieben werden, wodurch Sie sich zum Zeitpunkt der Programmdurchführung eine Auswahlmöglichkeit schaffen.

Sie können Ihr Programm mit dem LLIST-Kommando auflisten lassen (Einzelheiten siehe Kapitel 8). Wenn Sie LLIST ohne Zeilennummern verwenden, werden alle in dem Moment im Speicher befindlichen Programme nach der numerischen Reihenfolge ihrer Zeilennummern ausgegeben. Die auszugebenden Zeilen lassen sich begrenzen, indem Sie die entsprechende Spanne mit dem LLIST-Kommando eingeben. Wenn Programmzeilen länger als 24 Zeichen sind, werden für den Ausdruck zwei oder mehr Zeilen benutzt. Die zweite und alle folgenden Zeilen werden um vier oder sechs Zeichen eingerückt, so daß die jeweilige Zeilennummer eindeutig zu identifizieren ist. (Von Zeile 1 bis 999: Einrückung um 4 Zeichen, ab Zeile 999 Einrückung um 6 Zeichen.)

Achtung:

* Falls eine Fehlermeldung ERROR 8 infolge eines Papierstaus auftritt, reißen Sie das Papier ab und ziehen Sie das verbliebene Papier vollständig aus dem Drucker. Drücken Sie dann die [CL] -Taste, um die Fehlermeldung zu löschen.

* Wenn der Drucker starken externen elektrischen Störfeldern ausgesetzt ist, können eventuell falsche Zahlen willkürlich ausgedruckt werden. In diesem Fal drücken Sie die [BRK] -Taste, um den Druckvorgang zu unterbrechen. SchaltenSie die Stromversorgung des CE-126P ab und wieder an, drücken Sie dann die [CL] -Taste. Durch Druck auf die [CL] -Taste initialisieren Sie den Drucker.

* Wenn der CE-126P nicht in Betrieb ist, schalten Sie bitte den Drucker ab, um die Batterien zu schonen.

DER GEBRAUCH DES CASSETTEN-INTERFACE

Mit Hilfe dieses Cassetten-Interface können Sie Daten und Programme aus dem Computer auf Tonbandcassetten abspeichern (natürlich brauchen Sie dazu auch einen Cassettenrecorder wie das von uns zu diesem Taschencomputer-System gelieferte Modell CE-152). Sind die Daten und Programme einmal auf Band gespeichert, können Sie sie durch einen einfachen Ablauf wieder in den Computer einladen.

DAS ANSCHLIESSEN DES CE-126P AN EINEN CASSETTENRECORDER

Hier sind nur drei Verbindungen zu schaffen:

1. Stecken Sie den roten Klinkenstecker in die MICrofonbuchse des Cassettenrecorders.
2. Stecken Sie den grauen Klinkenstecker in die Kopfhörerbuchse am Cassettenrecorder.
3. Stecken Sie den schwarzen Stecker in die REMote-Buchse am Cassettenrecorder.

CASSETTENRECORDER

Wir empfehlen Ihnen, für Ihr Taschencomputer-System das Zusatzgerät CE-152 zu benutzen. Dieses Gerät ist auf den PC 1260/1261 zugeschnitten und nimmt Daten und Programme über das CE-126P Interface auf. Jedes abgespeicherte Programm kann wieder aufgesucht und in den PC 1260/1261 eingeladen werden.

Wenn Sie einen anderen Recorder benutzen als die Option CE-152, muß dieser den folgenden Mindestanforderungen entsprechen, um über das Interface CE-126P angeschlossen werden zu können:

Anforderungen

1. Recordertyp	Jedes Cassettengerät, ob mit Standard- oder Mikrocassetten, kann benutzt werden, sofern die unten aufgeführten Bedingungen erfüllt sind.
2. Aufnahmeanschluß	Der Recorder sollte über einen Klinkensteckeranschluß (2,5 mm) mit der Bezeichnung MIC verfügen. Benutzen Sie niemals den AUX-Anschluß.
3. Aufnahmeimpedanz	Der Aufnahmeanschluß sollte eine niedrige Impedanz besitzen (200 - 1.000 Ohm).

4. Minimales Aufnahmeniveau	Unter 3mV oder -50dB.
5. Wiedergabeanschluß	Sollte ein Klinkensteckeranschluß (2,5 mm) mit der Bezeichenung "EXT" (EXTerner Lautsprecher), "MONITOR", "EAR" (Kopfhörer) sein.
6. Wiedergabeimpedanz	Sollte unter 10 Ohm liegen.
7. Wiedergabeniveau	Sollte über 1V liegen (maximale Wiedergabe über 100 mW).
8. Verzerrung	Sollte innerhalb 15 %, innerhalb eines Bereiches zwischen 2 und 4 KHz liegen.
9. Gleichlaufschwankung	maximal 0.3 % (W.R.M.S.)

* Falls der mit dem CE-126P Interface gelieferte Klinkenstecker nicht zu dem Eingang an Ihrem Gerät paßt, können Sie im Fachhandel entsprechende Adapter bekommen.

Anmerkung: Einige Cassettengeräte können aufgrund spezieller Abweichungen von den Spezifikationen nicht anzuschließen sein. Weiterhin können mit Geräten, die infolge langjährigen Gebrauchs ihre elektrischen Eigenschaften verändert haben (starke Verzerrung, erhöhtes Rauschen, Spannungsabfall), unbefriedigende Ergebnisse erzielt werden.

DAS ARBEITEN MIT CASSETTEN-INTERFACE UND RECORDER ABSPEICHERN AUF MAGNETBAND

(Siehe auch Anmerkungen zum Tonband.)

1. Schalten Sie den REMOTE-Schalter des CE-126P auf OFF.

2. Geben Sie ein Programm oder Daten in den Computer ein.

3. Legen Sie die Cassette in den Recorder ein. Bestimmen Sie die Stelle auf dem Band, an der Sie das Programm abspeichern wollen.

   * Wenn Sie ein leeres Band benutzen, achten Sie darauf, daß Sie den Leervorspann aus nicht-magnetisierter Folie nicht bespielen.
   * Wenn Sie ein bereits teilweise bespieltes Band benutzen, suchen Sie eine Stelle, an der noch keine Daten gespeichert sind!

4. Stecken Sie den roten Stecker des Interface in die MICrofonbuchse des Cassettenrecorders und den schwarzen Stecker in die REM-Buchse.

5. Bringen Sie den REMOTE-Schalter in die Stellung ON.

6. Zur Aufnahme drücken Sie nun am Cassettenrecorder die Aufnahme- und die PLAY-Taste gleichzeitig.

7. Geben Sie die Aufnahmeinstruktionen ein (CSAVE-Kommando, PRINT#-Kommando und drücken Sie die [ENTER] -Taste).

   Setzen Sie die Einheit zunächst in den RUN- oder PRO-Modus und drücken Sie die folgenden Tasten:

   C S A V E [SHIFT]["] dateiname [SHIFT]["] [ENTER]

   (Um den Inhalt des Datenspeichers auf Band zu überspielen, drücken Sie die folgenden Tasten:

   P R I N T [SHIFT][#][ENTER])
   z. B. C S A V E [SHIFT]["]AA[SHIFT]["] [ENTER]

   Wenn Sie die [ENTER] -Taste drücken, wird das Band zu laufen beginnen, wobei ein etwa 8 Sekunden langes Stück unbespielt bleibt. (Ein Piepton ist zu hören.) Anschließend werden der Dateiname und sein Inhalt aufgenommen.

8) Wenn die Aufnahme vollständig ist, erscheint das Bereitschaftssymbol in der Anzeige (>) und der Recorder hält automatisch an. Nun haben Sie Ihr Programm im Cassettenrecorder abgespeichert. Es befindet sich jedoch trotzdem im Computer.

   Wenn Daten über einen Programmbefehl automatisch abgespeichert werden sollen (PRINT# -Befehl, nicht bei manueller Bedienung), befolgen Sie erst die Schritte 1 bis 6, ehe Sie das Programm ablaufen lassen.

   Als Hilfe beim Auffinden bestimmter Programme auf Band benutzen Sie bitte das Bandzählwerk des Recorders.

VERGLEICHEN VON SPEICHER- UND BANDINHALTEN

(Siehe auch Anmerkungen zum Tonband.)

Nachdem Sie ein Programm von Band geladen bzw. auf Band abgespeichert haben, können Sie überprüfen, ob das Programm im Computer identisch mit dem auf Band ist (und so sicher gehen, daß alles in Ordnung ist, ehe Sie mit dem Programmieren oder der Programmdurchführung fortfahren).

1. Schalten Sie den REMOTE-Schalter auf OFF.

2. Bewegen Sie das Band mit den Vor- und Rücklauftasten an die Stelle kurz vor Beginn des zu prüfenden Programms.

3. Stecken Sie den grauen Stecker in die Kopfhörerbuchse und den schwarzen Stecker in die REMote-Buchse.

4. Schalten Sie den REMOTE-Schalter auf ON.

5. Drücken Sie die PLAY-Taste des Recorders.

6. Geben Sie ein CLOAD?-Kommando ein und beginnen Sie den Prüfvorgang mit der [ENTER] -Taste. Gehen Sie dabei folgendermaßen vor:

   Setzen Sie den RUN- oder PRO-Modus.

   Geben Sie folgende Tastenfolge ein:

   C L O A D [SHIFT] ? [SHIFT] ["] AA [SHIFT] ["] [ENTER]

   ↑ Der Dateiname, den Sie vorher gewählt haben.

   Der Taschencomputer wird nun automatisch nach dem vorgegebenen Dateinamen suchen und wird den Inhalt des Tonbandes mit dem Speicherinhalt vergleichen.

   Während des Vergleichsvorganges erscheint das Zeichen "*" an der äußersten rechten Stelle der unteren Display-Zeile. Das Zeichen "*" erlischt, wenn der Vergleich beendet ist. Solange der Computer nach dem Dateinamen sucht, wird das "*"-Zeichen nicht in der Anzeige ausgegeben, da der eigentliche Vergleich noch nicht begonnen hat. (Das gleiche geschieht, wenn das erste Programm ohne Dateinamen gelesen wird.)

   Werden die Programme als übereinstimmend erkannt, erscheint das Bereitschaftssymbol (>) in der Anzeige des Taschencomputers.

   Decken sich die Programme nicht, wird der Vergleich unterbrochen und die Meldung ERROR 8 ausgegeben. Wenn dies geschieht, versuchen Sie es bitte noch einmal.

EINLADEN VON MAGNETBAND

(Siehe auch Anmerkungen zum Tonband.)

Um Programme oder Daten vom Magnetband in den Computer einzuladen, gehen Sie bitte folgendermaßen vor:

1. Schalten Sie den REMOTE-Schalter auf OFF.

2. Legen Sie das Band in den Recorder ein. Bringen Sie das Magnetband mit Hilfe der Vor- bzw. Rücklauftasten kurz vor die Stelle, an der das einzulesende Programm beginnt.

3. Stecken Sie den grauen Stecker in die Kopfhörerbuchse des Recorders und den schwarzen Stecker in die REM-Buchse. (Falls Sie einen Recorder benutzen, der keinen REM-Eingang hat, drücken Sie für eine kurze Unterbrechung die PAUSE-Taste.)

4. Schalten Sie den REMOTE-Schalter wieder auf ON.

5. Drücken Sie den PLAY-Schalter des Recorders, um die Einheit in den Playback-Modus zu versetzen. Stellen Sie VOLUME auf mittlere oder maximale Lautstärke ein. Drehen Sie die Höhen maximal auf.

6. Geben Sie die Übertragungsanweisungen (CLOAD-Kommando, INPUT#-Kommando) ein und starten Sie die Ausführung mit [ENTER] .

   Setzen Sie die Einheit in den RUN-Modus, danach betätigen Sie die folgenden Tasten:

   C L O A D [SHIFT] ["] Dateiname [SHIFT] ["] [ENTER]

   (Um den Inhalt den Datenspeichers zu laden, drücken Sie:

   I N P U T [SHIFT] [#] [ENTER])

   Z. B.:

   C L O A D [SHIFT] ["] AA [SHIFT] ["] [ENTER]

   Der vorgegebene Dateiname wird nun automatisch gesucht und sein Inhalt wird in den Taschencomputer eingeladen. Das Zeichen "*" erscheint, solange der Ladevorgang abläuft. Das gleiche geschieht, wenn das erste Programm ohne Dateinamen gelesen wird.

   Das Zeichen "*" erlischt, wenn das Programm vollständig geladen ist.

7. Wenn das Programm übertragen worden ist, hält der Computer automatisch den Bandlauf an und das Bereitschaftssymbol (>) erscheint in der Anzeige.

   Um im Verlauf eines Programms Daten zu übertragen (INPUT#-Befehl), führen Sie erst die Schritte 1 bis 5 aus, ehe Sie das Programm starten.

Anmerkungen:

* Wenn Sie eine Fehlermeldung erhalten (ERROR 8), fangen Sie noch einmal von vorn an. Wiederholt sich die Fehlermeldung, verändern Sie vorsichtig die VOLUME-Einstellung.

* Wird keine Fehlermeldung ausgegeben und läuft das Band weiter (während der Computer die Anzeige BUSY ausgibt), verläuft die Übertragung fehlerhaft. Drücken Sie dann die [BRK] -Taste, um die Übertragung abzubrechen und wiederholen Sie die Schritte.

* Bleibt die Fehlermeldung bestehen oder läuft das Band auch nach mehreren Korrekturversuchen weiter, reinigen und entmagnetisieren Sie den Tonkopf des Recorders.

ANMERKUNGEN ZUM TONBAND

1. Benutzen Sie zur Datenübertragung oder zum Vergleich ausschließlich den Recorder, mit dem die Programme aufgezeichnet wurden. Die Verwendung eines anderes Gerätes kann möglicherweise die Datenübertragung oder den Programmvergleich ausschließen.

2. Benutzen Sie grundsätzlich Tonbandcassetten der höchsten Qualität, um Daten und Programme darauf abzuspeichern (Bandmaterial zum "Sparpeis" weist u. U. nicht die für digitale Aufzeichnungen erforderlichen Eigenschaften auf!).

3. Halten Sie die Tonköpfe und das Cassettenlaufwerk sauber - benutzen Sie eine Reinigungscassette zur Reinigung.

4. Lautstärke-Einstellung: mittlere oder maximale Lautstärke. Die Lautstärke-Einstellung kann von großer Bedeutung beim Lesen von Daten vom Band sein. Ändern Sie die Einstellung geringfügig, wenn dies erforderlich wird. Schon geringe Änderungen können zu hervorragenden Ergebnissen führen.

5. Achten Sie darauf, daß alle Verbindungen zwischen Computer und Cassetten-Interface sicher sind. Dasselbe gilt für die Verbindungen zwischen Interface und Recorder. Darüber hinaus müssen die Verbindungselemente sauber und staubfrei sein.

6. Sollte es beim Gebrauch des AC-Netzteils Probleme geben, speisen Sie den CE-126P und/oder den Recorder über Batterien (manchmal kommt durch die Benutzung des AC-Netzteils ein Summen zu den zu übertragenden Signalen, das die eigentliche Aufnahme stört.)

* Wenn Sie das AC-Netzteil an den CE-126P anschließen, schalten Sie die Stromversorgung des CE-126P ab und stellen Sie die Verbindung erst dann her.

7. Höhenregler: auf maximale Höhen einstellen.

8. Wenn Sie Programme oder Daten auf einem bereits gebrauchten Band abspeichern wollen, löschen Sie diesen Teil vor der Eingabe und geben Sie dann den Speicherbefehl. (Vergewissern Sie sich, daß das alte Programm restlos gelöscht ist.)

KAPITEL 8

BASIC REFERENZ TEIL

Das folgende Kapitel ist in drei Teile untergliedert:

KOMMANDOS — Instruktionen an den Rechner, die außerhalb eines Programms gegeben werden, um Arbeitsbedingungen, Zusatzschaltungen und Programmkontrollen zu bestimmen.

BEFEHLE — Kommando- und Befehlswörter, die gebraucht werden, um ein Programm zu erstellen.

FUNKTIONEN — Spezielle BASIC-Operatoren, um Variable umzuwandeln.

Die Kommandos und die Befehle sind alphabetisch geordnet. Um ein Befehl oder ein Kommando schnell finden zu können, beginnt jede Eingabe auf einer neuen Seite. Der Inhalt jedes Abschnitts ist in den Tabellen auf dieser und den folgenden Seiten wiedergegeben. Anhand dieser Tabellen können Sie ein Schlüsselwort schnell der entsprechenden Kategorie zuordnen und nachschlagen. Funktionen sind in vier Gruppen und in diesen alphabetisch geordnet.

KOMMANDOS

Programmkontrolle

CONT b	NEW
GOTO b	RUN

Cassettenkontrolle

CLOAD	INPUT# b
CLOAD?	MERGE
CSAVE	PRINT# b

Debug-Kontrolle

LIST	TROFF b
LLIST	TRON b

Variablenkontrolle

CLEAR b	
DIM b	

Winkelmodus

DEGREE b	RADIAN b
GRAD b	

Andere

BEEP b	USING b
PASS b	WAIT b
RANDOM b	

Mit einem "b" versehene Kommandos sind auch BASIC-Befehle. Sie werden nicht im Kommando-Teil dieses Kapitels, sondern im Befehls-Teil beschrieben, da ihre Wirkung als Kommando und als Befehl identisch ist.

BEFEHLE

Kontrollen und Verzweigungen

CHAIN
END
FOR...TO...STEP
GOSUB
GOTO
IF...THEN
NEXT
ON...GOSUB
ON...GOTO
RETURN
STOP

Eingabe und Ausgabe

AREAD
CSAVE
CURSOR
DATA
INPUT
INPUT=
LPRINT
PAUSE
PRINT
PRINT=
READ
RESTORE
USING
WAIT

Zuweisung und Deklarationen

CLEAR
DIM
LET

Andere

BEEP
CLS
DEGREE
GRAD
RADIAN
RANDOM
REM
TROFF
TRON

FUNKTIONEN

Pseudovariable

INKEY$
MEM
PI

String-Funktionen

ASC
CHR$
LEFT$
LEN
MID$
RIGHT$
STR$
VAL

Numerische Funktionen

ABS
ACS
ASN
ATN
COS
DEG
DMS
EXP
INT
LN
LOG
RND
SGN
SIN
SQR
TAN

Kommando-Funktionen

EQU=
LIST=
LLIST=
MEM=
NEW=

KOMMANDOS

CLOAD

1. CLOAD
2. CLOAD "dateiname"

Abkürzung: CLO., CLOA.
Vergleiche: CLOAD?, CSAVE, MERGE, PASS

Wirkung:
Mit dem CLOAD-Kommando werden Programme von der Cassette in den Rechner geladen. Das Kommando kann nur in Verbindung mit dem Drucker/Mikrocassettenrecorder CE-125 benutzt werden.

Anwendung:
Die erste Form des CLOAD-Kommandos löscht den Speicher des Rechners und lädt das erste Programm auf der eingelegten Cassette.

Die zweite Form des Kommandos löscht den Speicher und lädt das mit "dateiname" angegebene Programm von der Cassette.

Ist der PC 1260/1261 im PROgramm- oder RUN-Modus, so wird das Programm von der Cassette in den Programmspeicher geladen. Ist der PC 1260/1261 im ReSerVe-MODUS, so werden die Daten in den Reservespeicher geladen. Sie müssen darauf achten, daß Sie keine Programme in den Reservespeicher und keine Reservedaten in den Programmspeicher laden.

Beispiele:

CLOAD	Lädt das erste Programm von der Cassette.
CLOAD "PRO3"	Sucht auf der eingelegten Cassette das Programm PRO3 und lädt es ein.

Bemerkungen:
1. Der Computer kann nicht entscheiden, ob die Daten für den Programm- oder den Reservespeicher gedacht sind. Werden Daten in den falschen Speicherbereich geladen, kann der Rechner nicht arbeiten. Sollte dieser Fehler gemacht worden sein, müssen Sie den RESET-Knopf an der Rückseite des Rechners drücken.
2. Wird die angegebene Datei nicht gefunden, sucht der Rechner weiter nach der Datei, selbst wenn die Cassette abgelaufen ist. Unterbrechen Sie in diesem Fall die Suche mit der [BRK] -Taste. Das bezieht sich auch auf die Kommandos MERGE, CHAIN, CLOAD? und INPUT#, die später beschrieben werden.
3. Tritt während der Ausführung der Kommandos CLOAD oder CHAIN (wird später beschrieben) ein Fehler auf, so ist das im Speicher vorhandene Programm defekt.

CLOAD?

1. CLOAD?
2. CLOAD? "dateiname"

Abkürzung: CLO.?, CLOA.?
Vergleiche: CLOAD, CSAVE, MERGE, PASS

Wirkung:
Mit dem Kommando CLOAD? vergleichen Sie das Programm im Speicher des Rechners mit einem auf Cassette gespeicherten Programm. Es kann nur in Verbindung mit dem Drucker/Mikrocassettenrecorder CE-125 verwendet werden.

Anwendung:
Die erste Form des CLOAD?-Kommados vergleicht den Speicherinhalt mit dem ersten auf der Cassette gefundenen Programm.

Die zweite Form des Kommandos CLOAD? sucht nach dem mit "dateiname" angegebenen Programm und vergleicht es dann mit dem Speicherinhalt.

Beispiele:

CLOAD?	Vergleicht den Speicherinhalt mit dem ersten auf der der Cassette gefundenen Programm.
CLOAD? "PRO3"	Sucht auf der eingelegten Cassette nach dem Programm PRO3 und vergleicht es dann mit dem Speicherinhalt.

CONT

1. CONT

Abkürzung: C., CO., CON.
Vergleiche: RUN, STOP-Befehl

Wirkung:
Mit dem CONT-Kommando setzen Sie die Ausführung eines unterbrochenen Programms fort.

Anwendung:
Wurde die Ausführung eines Programms mit dem Befehl STOP oder mit der Taste [BRK] unterbrochen, so kann es mit dem Kommando CONT fortgesetzt werden.

Beispiel:

CONT Setzt die Ausführung eines unterbrochenen Programms fort.

CSAVE

1. CSAVE
2. CSAVE "dateiname"
3. CSAVE, "passwort"
4. CSAVE "dateiname", "passwort"

Abkürzung: CS., CSA., CSAV.
Vergleiche: CLOAD, CLOAD?, MERGE, PASS

Wirkung:
Mit diesem Kommando wird ein Programm aus dem Speicher des Rechners auf der Cassette abgespeichert. Es kann nur in Verbindung mit dem Drucker/Mikrocassettenrecorder CE-125 benutzt werden.

Anwendung:
Die erste Form des Kommandos CSAVE speichert das Programm im Speicher des Rechners ohne speziellen Dateinamen auf der Cassette ab.
Die zweite Form des Kommandos weist dem Programm vor Abspeicherung den angegebenen Dateinamen zu und speichert es unter diesem Namen auf der Cassette ab.

Die dritte Form des Kommandos speichert das Programm ohne Dateinamen, aber mit einem Passwort auf der Cassette ab. Programme, die mit einem Passwort geschützt sind, können von jedem geladen und ausgeführt werden. Gelistet oder verändert werden kann es aber nur dann, wenn das korrekte Passwort eingegeben wird (näheres unter der Beschreibung des PASS-Kommandos).

Mit der vierten Form des CSAVE-Kommandos wird dem Programm sowohl ein Dateiname als auch ein Passwort zugewiesen.

Im PROgramm- oder RUN-Modus wird der Inhalt des Programmspeichers, im ReSerVe-Modus der Inhalt des Reservespeichers auf Cassette abgespeichert.

Beispiel:

CSAVE "PRO3", "GEHEIM" Speichert das präsente Programm unter dem Dateinamen PRO3 mit dem Passwort GEHEIM auf Cassette ab.

GOTO

1. GOTO ausdruck

Abkürzung: G., GO., GOT.
Vergleiche: RUN

Wirkung:
Mit dem GOTO-Kommando wird die Ausführung eines Programms begonnen.

Anwendung:
Das GOTO-Kommando kann genauso wie das RUN-Kommando benutzt werden. Die Ausführung eines Programms wird von der durch ausdruck angegebenen Zeilennummer gestartet.

GOTO unterscheidet sich in sechs Punkten vom RUN-Kommando:

1. Der Wert des WAIT-Intervalles wird nicht zurückgesetzt.
2. Wurde die Anzeige durch einen USING-Befehl formatiert, wird sie nicht gelöscht.
3. Werte von Variablen und Feldern (Arrays) bleiben erhalten.
4. PRINT=LPRINT-Status wird nicht zurückgesetzt.
5. Der READ-Zeiger wird nicht zurückgesetzt.
6. Die Cursor-Spezifikationen bleiben erhalten.

Beispiel:

GOTO 100 Die Ausführung eines Programmes wird bei Zeile 100 gestartet.

LIST

1. LIST
2. LIST ausdruck

Abkürzung: L., LI., LIS.
Vergleiche: LLIST

Wirkung:
Mit dem LIST-Kommando wird ein Programm zur Anzeige gebracht.

Anwendung:
Das LIST-Kommando kann nur im PROgramm-Modus benutzt werden. Die erste Form des Kommandos beginnt die Auflistung des Programms mit der kleinsten Zeilennummer.

Die zweite Form des Kommandos bringt die Programmzeile, die gleich oder die nächstgrößere zum Wert des eingegebenen Ausdrucks ist, zur Anzeige. Mit den Hoch- und Runter-Pfeilen können dann die Vorgänger- bzw. Nachfolgerzeilen zur Anzeige gebracht werden. Sind mit dem MERGE-Kommando mehrere Programme in den Speicher geladen, so wird mit dem LIST-Kommando das letzte eingeladene Programm gelistet (näheres in der Beschreibung des MERGE-Kommandos).

Wird das Kommando in der Form

```
LIST "label"
```

benutzt und das angegebene Label im zuletzt geladenenen Programm nicht gefunden, so sucht das Kommando das Label in den anderen Programmen im Speicher. Die dazugehörige Zeile wird dann gelistet.

Bei einem durch ein Passwort geschützten Programm wird das LIST-Kommando nicht ausgeführt.

Beispiel:

```
LIST 100        Die Zeile 100 wird angezeigt.
```

LLIST

1. LLIST
2. LLIST ausdruck
3. LLIST ausdruck1,ausdruck2
4. LLIST ausdruck.
5. LLIST,ausdruck

Abkürzung: LL., LLI., LLIS.
Vergleiche: LIST

Wirkung:
Mit dem LLIST-Kommando kann ein Programm auf dem Drucker des Drucker/Mikrocassettenrecorders CE-125 ausgegeben werden.

Anwendung:
Das LLIST-Kommando kann im PROgramm-Modus und im RUN-Modus benutzt werden.
Die erste Form druckt das gesamte Programm im Speicher aus. Die zweite Form druckt nur die mit ausdruck angegebene Zeile aus. Die dritte Form des Kommandos drukct das Programm a der durch den ausdruck1 angegebenen Zeile oder der nächstgrößeren, bis zu der druch den ausdruck2 angegebenen Zeile oder der nächstgrößeren aus. Zwischen den beiden angegebenen Zeilen müssen mindestens zwei Zeilen liegen. Die vierte Form listet das Programm ab der durch ausdruck eingegebenen Zeile bis zum Ende. Die fünfte Form des Kommandos LLIST druckt die Programmzeilen bis zur eingegebenen Zeile (inklusive) aus.

Wenn mit dem MERGE-Kommando mehrere Programme eingeladen wurden, so arbeitet das LLIST-Kommando mit dem zuletzt geladenen Programm.

Wird das Kommando in der Form

6. LLIST "label"
oder
7. LLIST,"label"

gegeben und das angegebene Label im zuletzt geladenenen Programm nicht gefunden, so durchsucht das Kommando die anderen Programme. Nach Form sechs wird dann die Zeile mit dem angegebenen Label ausgedruckt. Mit Form sieben wird die Zeile mit dem Label und das folgende Programm ausgegeben.

Ein gesetztes Passwort unterdrückt das LLIST-Kommando.

Beispiel:

LLIST 100,200 Die Programmzeilen zwischen 100 und 200 werden ausgedruckt.

MERGE

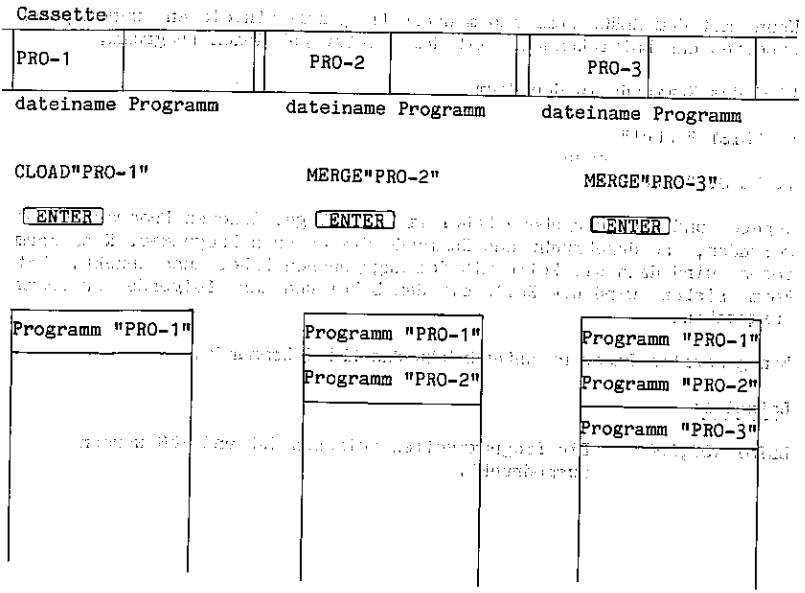
1. MERGE
2. MERGE "dateiname"

Abkürzung: MER., MERG.
Vergleiche: CLOAD

Wirkung:
Mit dem MERGE-Kommando können Programme von der Cassette an bereits im Speicher vorhandene angeknüpft werden.

Anwendung:
Das Kommando kann nur im PROgramm- und im RUN-Modus benutzt werden. Mit dem MERGE-Kommando wird, ohne daß das bereits im Speicher geladene Programm gelöscht wird, ein weiteres Programm von der Cassette in den Speicher des Rechners geladen. Auf diese Weise können völlig verschiedene Programme gleichzeitig im Speicher geladen sein.

Beispiele:
Nehmen wir an, wir haben drei Programme mit den Dateinamen PRO1, PRO2 und PRO3 auf Cassette abgespeichert. Jetzt kann das Programm PRO1 mit dem Kommando CLOAD eingeladen werden, die Programme PRO2 und PRO3 mit dem MERGE-Kommando. Die Abbildung auf dieser Seite verdeutlicht, wie die Programme gespeichert werden.



Wenn Programme mit dem MERGE-Kommando geladen wurden, kann es vorkommen, daß mehrere Programme die gleichen Zeilennummern haben.

Die Kommandos RUN und GOTO arbeiten dann nur mit dem zuletzt mit MERGE geladenen Programm. Da die vorher geladenenen Programme nicht ausgeführt werden können, müssen sie durch Tasten, z. B. A, S, D und so weiter vorher definiert werden. Solche Programme werden "definierte Programme" genannt. Näheres dazu im Kapitel "Definierte Programme".

Das gleiche gilt für die Editierungsbefehle und Kommandos. Nur das zuletzt geladene Programm kann editiert werden. Die vorher geladenenen Programme können nur dann editiert werden, wenn sie definiert wurden.

VERKNÜPFUNG PASSWORT-GESCHÜTZTER PROGRAMME

Wollen Sie ein Passwort-geschütztes Programm mit dem MERGE-Kommando laden, so müssen wir zwei Fälle unterscheiden:

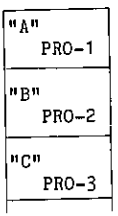
1. Das Programm im Speicher ist geschützt.
2. Das Programm im Speicher ist nicht geschützt.

Im ersten Fall kann kein Programm mit MERGE geladen werden. Im zweiten Fall werden alle Programme im Speicher durch das neu geladene, geschützte Programm ebenfalls zu Passwort-geschützten Programmen.

Sind im Speicher geschützte Programme und wird ein ungeschütztes Programm mit MERGE geladen, so wird dieses neu geladene Programm ebenfalls zu einem Passwort-geschützten Programm.

AUSFÜHRUNG MIT MERGE GELADENER PROGRAMME

Die Abbildung zeigt die Speicherorganisation, nachdem PRO1 mit CLOAD und PRO2 und PRO3 mit MERGE geladen wurden. Wenn Sie RUN oder GOTO benutzen, um ein Programm zu starten, so wird immer PRO3 ausgeführt.



Die Programme PRO1 und PRO2 können nur dann gestartet werden, wenn ein Label gesetzt wurde und jetzt im RUN- oder GOTO-Kommando benutzt wird:

```
RUN "label"
GOTO "label"
```

Eine weitere Möglichkeit ist, das Programm mit einer definierten Taste aufzurufen. Der Rechner sucht dann nach dem im Kommando oder durch die Taste gegebenen Label, und startet das dazugehörende Programm der Zeile mit dem Label.

Bitte beachten Sie, daß die Programme mit verschiedenen Labels versehen werden. Der Rechner durchsucht die Programme nach dem Label in der Reihenfolge PRO3, PRO1 und PRO2. Ist also ein Label in PRO3 gleich dem Label von PRO1, so wird immer PRO3 ausgeführt, wenn das Label aufgerufen wird.

Das gleiche geschieht, wenn das LIST-Kommando gegeben wird. Wird das Kommando

```
LLIST "A","C" ENTER
```

gegeben und steht das Label "C" vor "A", so wird das Kommando nicht ausgeführt, und eine Fehlermeldung wird ausgegeben.

NEW

1. NEW

Vergleiche: NEW=

Wirkung:
Das NEW-Kommando löscht Programme im Speicher bzw. Daten im Reservespeicher.

Anwendung:
Wird es im PROgramm-Modus benutzt, so werden alle Programme und Daten gelöscht.

Im ReSerVe-Modus wird der Reservespeicher gelöscht.

Das NEW-Kommando ist im RUN-Modus nicht definiert. Sie bekommen die Fehlermeldung ERROR 9.

Beispiel:

NEW Löscht den entsprechenden Speicherbereich.

PASS

1. PASS "zeichenkette"

Abkürzung: PA., PAS.
Vergleiche: CSAVE, CLOAD

Wirkung:
Mit dem Kommando PASS können Passwörter gesetzt und gelöscht werden.

Anwendung:
Mit einem Passwort schützen Sie Ihr Programm vor Einsichtnahme oder Modifikation durch andere Anwender. Ein Passwort besteht aus einer Zeichenkette bis zu sieben Zeichen. Die erlaubten Zeichen für das Passwort sind alle Buchstaben des Alphabets und folgende Sonderzeichen:

```
! ≠ $ % & ( ) *
+ - / . , : ; <
= > ? @ ^
```

sowie das Wurzelzeichen und Pi.

Wurde das Kommando PASS gegeben, so ist das Programm im Speicher geschützt. Das bedeutet, daß das Programm nicht mehr editiert werden kann. Es kann nicht auf Cassette abgespeichert oder mit LIST oder LLIST ausgegeben werden. Weiterhin können weder Zeilen hinzugefügt noch gelöscht werden. Die einzige Möglichkeit, das Passwort auszuschalten, ist ein weiteres PASS-Kommando mit dem gleichen Passwort zu geben oder das NEW-Kommando zu geben, womit das präsente Programm allerdings gelöscht wird.

Beispiel:

PASS "GEHEIM" Setzt für alle Programme im Speicher das Passwort "GEHEIM".

RUN

1. RUN
2. RUN zeilennummer

Abkürzung: R., RU.
Vergleiche: GOTO

Wirkung:
Mit dem RUN-Kommando wird das Programm im Speicher gestartet.

Anwendung:
Die erste Form des Kommandos startet das Programm im Speicher mit der Zeile, die die niedrigste Zeilennummer hat.

Die zweite Form startet das Programm mit der angegebenen Zeilennummer. Die Kommandos RUN und GOTO unterscheiden sich in sechs Punkten. Für das Kommando RUN gilt folgendes:

1. Der Wert des WAIT-Intervalls wird zurückgesetzt.
2. Eine durch einen USING-Befehl erzeugte Formatierung der Anzeige wird gelöscht.
3. Variable und Felder (Arrays) werden gelöscht.
4. PRINT=PRINT-Status wird gesetzt.
5. Der READ-Zeiger wird auf den Anfang der ersten DATA-Zeile gesetzt.
6. Die Cursor-Spezifikationen werden gelöscht.

Die Ausführung eines Programms mit GOTO und mit der [DEF] -Taste sind gleich. Alle drei Formen des Programmstarts löschen FOR/NEXT- und GOTO-Gruppen.

Beispiel:

```
RUN 100          Startet ein Programm mit der Zeile 100.
```

BEFEHLE

AREAD

1. AREAD variablenname

Abkürzung: A., AR., ARE., AREA.
Vergleiche: Befehl INPUT und die Beschreibung der DEF -Taste in Kapitel 6.

Wirkung:
Wenn einem Programm ein Label (ein "Name") in Form eines Buchstabens zugewiesen wurde, so daß es mit der DEF -Taste gestartet werden kann, kann der AREAD-Befehl benutzt werden, um einen einzelnen Startwert einzugeben, ohne den INPUT-Befehl zu setzen. Der Befehl AREAD muß in der ersten Zeile nach der Zeile mit dem Label gesetzt sein. Taucht er an anderer Stelle im Programm auf, so wird er ignoriert. Es können sowohl Zeichenketten (Strings) als auch numerische Werte eingegeben werden, aber immer nur ein Wert pro Programm.

Erstellen Sie Ihr Programm im PROgramm-Modus. Der AREAD-Befehl kommt dann im RUN-Modus zum Tragen. Geben Sie erst den gewünschten Wert und dann drücken Sie erst DEF , gefolgt von dem Label, das Sie gesetzt haben. Wird ein String benutzt, so muß dieser nicht in Anführungsstriche gesetzt werden.

Beispiele:

```
10 "X": AREAD N          Erstellen Sie sich dieses Beispiel-Programm
20 PRINT N^2             im PROgramm-Modus.
30 END
```

Wechseln Sie jetzt in den RUN-Modus.

Die Eingabe

7 DEF X

gibt das Quadrat der Zahl "7", also "49", aus.

Wichtig:

1. Wird das Eingabe-PROMPT (>) zum Beginn der Programmausführung angezeigt, so ist die benutzte Variable gelöscht.

2. Wird zu Beginn des Programms mit dem PRINT-Befehl etwas angezeigt, so wird folgendes gespeichert:

   Folgendes Programm soll ausgeführt werden:

```
10 "A":PRINT "ABC","DEFG"      Erstellung des Programms im
20 "S":AREAD A$:PRINT A$       PROgramm-Modus.
```

Wechsel in den RUN-Modus:

EINGABE	ANZEIGE
DEF A	ABC DEFG
DEF S	DEFG

* Wird PRINT numerischer ausdruck,numerischer ausdruck,numerischer ausdruck, numerischer ausdruck

  oder

  PRINT string,string,string,string ausgegeben, so wird der zuletzt angezeigte Ausdruck gespeichert.

* Wird PRINT numerischer ausdruck;numerischer ausdruck;numerischer ausdruck;... ausgegeben, so wird der erste angezeigte Ausdruck gespeichert.

* Wird PRINT string;string;string;... ausgegeben, so wird der zuletzt ausgegebene Ausdruck gespeichert.

BEEP

1. BEEP ausdruck

Abkürzung: B., BE., BEE.

Wirkung:
Mit dem BEEP-Befehl wird ein Ton erzeugt.

Anwendung:
Der PC 1260/1261 erzeugt auf den BEEP-Befehl hin einen oder mehrere 4kHz-Töne. Die Anzahl der Töne wird mit "ausdruck" angegeben (positive Zahlen kleiner als 9.999999999 E 99). Nur der ganzzahlige Anteil des angegebenen Ausdrucks wird im BEEP-Befehl verwertet.

BEEP kann auch mit numerischen Variablen arbeiten oder als Kommando benutzt werden. In diesem Falle werden die Töne direkt nach dem [ENTER] erzeugt.

Beispiele:

```
10 A=5 : B$="9"
20 BEEP 3          Erzeugt 3 Töne.
30 BEEP A          Erzeugt 5 Töne.
40 BEEP (A+4)/2    Erzeugt 4 Töne.
50 BEEP B$         Nicht erlaubt, ERROR 9 wird ausgegeben.
60 BEEP -4         Es wird kein Ton erzeugt, aber auch keine Fehler-
                   meldung ausgegeben.
```

CHAIN

1. CHAIN
2. CHAIN ausdruck
3. CHAIN "dateiname"
4. CHAIN "dateiname",ausdruck

Abkürzung: CH., CHA., CHAI.
Vergleiche: CLOAD, CSAVE, RUN

Wirkung:
Mit dem CHAIN-Befehl wird ein auf Cassette gespeichertes Programm ausgeführt. Er kann nur in Verbindung mit dem Drucker/Mikrocassettenrecorder CE-125 benutzt werden.

Anwendung:
Um den CHAIN-Befehl benutzen zu können, muß mindestens ein Programm auf Cassette abgespeichert sein. Dieses Programm wird dann mit dem CHAIN-Befehl geladen und ausgeführt.

Die erste Form des Befehls lädt das erste gefundene Programm von Cassette und beginnt die Ausführung mit der niedrigsten Zeile dieses Programms. Der Effekt ist der gleiche wie mit dem CLOAD- und dem RUN-Kommando im RUN-Modus.

Die zweite Form lädt das erste gefundene Programm und beginnt die Ausführung mit der durch ausdruck angegebenen Zeile.

Die dritte Form von CHAIN sucht das mit dateiname angegebene Programm, lädt es und führt es aus, beginnend mit der kleinsten Zeilennummer dieses Programms.

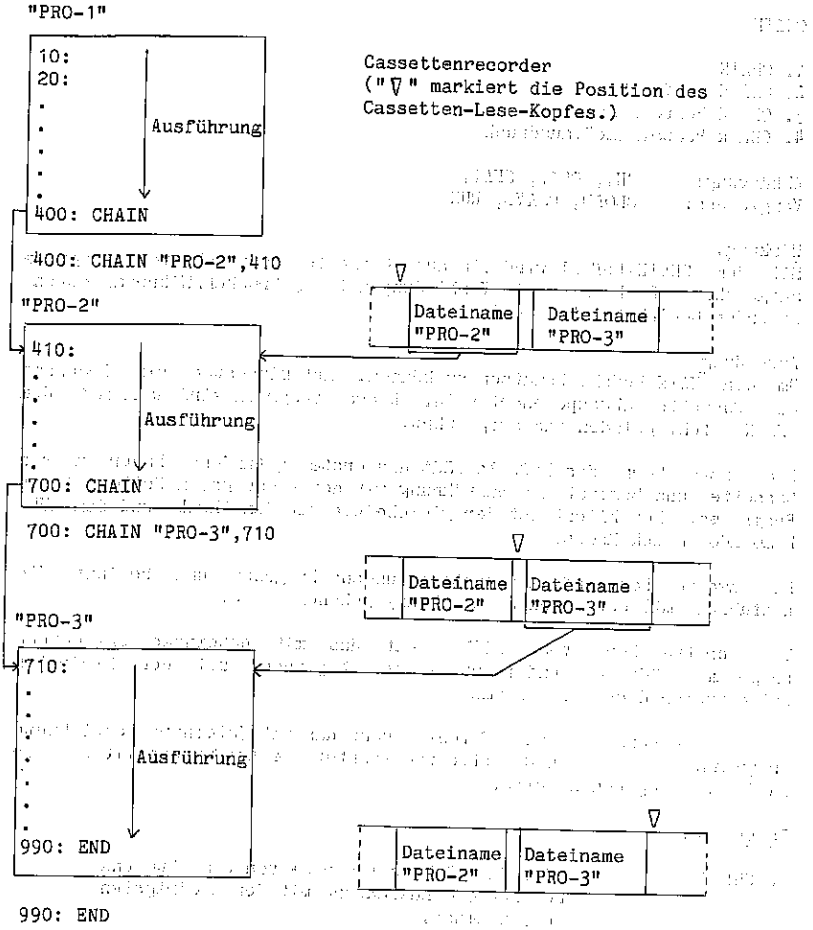
Die vierte Form des CHAIN-Befehls sucht das mit dateiname bezeichnete Programm, lädt es von Cassette und startet die Ausführung mit der in ausdruck angegebenen Zeile.

Beispiele:

```
10 CHAIN                Lädt das nächste Programm von Cassette und
                        beginnt die Ausführung mit der niedrigsten
                        Zeilennummer.

20 CHAIN "PRO2",480     Sucht auf der Cassette das Programm PRO2, lädt
                        es und startet es mit Zeile 480.
```

Nehmen Sie z. B. einmal an, daß Sie drei Programme mit dem Namen PRO1, PRO2 und PRO3 auf Cassette gespeichert haben. Die beiden ersten Programme enden mit einem CHAIN-Befehl. Das erste Programm laden Sie ganz normal mit dem CLOAD-Kommando.

Wenn Sie jetzt das Programm PRO1 ausführen, so lädt der Rechner am Ende des Programms das nächste Programm PRO2 automatisch nach und führt es aus, bis es auf den CHAIN-Befehl am Ende dieses Programms stößt. Jetzt lädt er das dritte Programm PRO3 und führt es aus.


"PRO-1"
10:
20:
Ausführung
400: CHAIN
400: CHAIN "PRO-2",410
"PRO-2"
410:
Ausführung
700: CHAIN
700: CHAIN "PRO-3",710
"PRO-3"
710:
Ausführung
990: END
990: END
Cassettenrecorder
("▽" markiert die Position des
Cassetten-Lese-Kopfes.)
Dateiname "PRO-2"
Dateiname "PRO-3"
Dateiname "PRO-2"
Dateiname "PRO-3"
Dateiname "PRO-2"
Dateiname "PRO-3"

CLEAR

1. CLEAR

Abkürzung: CL., CLE., CLEA.
Vergleiche: DIM

Wirkung:
Mit dem CLEAR-Befehl werden alle in einem Programm benutzten Variablen gelöscht und alle gesetzten Variablen auf Null bzw. auf leer gesetzt.

Anwendung:
Der CLEAR-Befehl setzt Speicherraum frei, der gebraucht wurde, um Werte von Variablen zu speichern. Dieser Befehl ist sehr hilfreich, wenn der Speicherbereich eingeschränkt ist und Variablen des ersten Teils eines Programms im zweiten Teil nicht mehr gebraucht werden. Wird CLEAR an den Anfang eines Programms gesetzt, so werden Speicherbereiche, die von den Variablen anderer Programme im Rechner belegt sind, frei gemacht.

CLEAR setzt die Speicherbereiche, die durch die Variablen A-Z, A$-Z$ oder A(1)-A(26) (ohne DIMensionierung) nicht frei, wenn diese fest vorgegeben wurden (siehe Kapitel 4). CLEAR setzt diese Variablen auf Null (numerische) bzw. auf leer (String).

Beispiele:

```
10 A=5: DIM C(5)
20 CLEAR
```

Setzt den C( ) zugewiesenen Speicher frei und A gleich Null.

CLS

1. CLS

Vergleiche: CURSOR

Wirkung:
Der CLS-Befehl löscht das Anzeigefeld.

Anwendung:
Der CLS-Befehl löscht die Anzeige und setzt das PROMPT, die Eingabe-Bereit-Anzeige, auf die 0-Position oben links, die auch Home-Position genannt wird.

Beispiel:

```
10:WAIT 30
20:INPUT A$
30:FOR B=0 TO 47
40:CLS
50:CURSOR B
60:PRINT A$
70:NEXT B
```

Dieses Beispiel-Programm bewegt das eingegebene Wort von der linken Seite der oberen Zeile zur rechten Seite der unteren Zeile. In jedem Durchgang der FOR/NEXT-Schleife wird die Anzeige mit dem CLS-Befehl gelöscht und die Cursor-Position mit dem CURSOR-Befehl verschoben. Wenn Sie Zeile 40 löschen, sehen Sie die Funktion des CLS-Befehls.

CURSOR

1. CURSOR ausdruck

Abkürzung: CU., CUR., CURS., CURSO.
Vergleiche: CLS, INPUT, PRINT, PAUSE

Wirkung:
Der CURSOR-Befehl setzt die Wiedergabe-Start-Position.

Anwendung:
Der ausdruck im CURSOR-Befehl gibt die Position an, in der ein Wert mit dem PRINT-Befehl wiedergegeben werden soll. Der Wert des Ausdrucks muß zwischen 0 und 47 inklusiv liegen, sonst erhalten Sie die Fehlermeldung ERROR 9. Die Wiedergabe-Positionen ersehen Sie aus der Abbildung. Die obere Zeile hat die Positionen 0 bis 23, die untere die Positionen 24 bis 47.

```
0 1 2 3 4 5 6 7 8 9 1011 . . . . . 2223

242526 . . . . . 4647
```

Beispiele:

```
10:CURSOR 4: PRINT "A"
20:CURSOR 22: PRINT "FG"
30:CURSOR 12: PRINT "BCDE"
40:CURSOR 34: PRINT "HIJKL"
```

Wenn Sie das Programm starten und die [ENTER] -Taste drücken, erhalten Sie dieses Bild:

```
0 1 2 3 4 . . . . . 12 . . . . . 2223
┌──────────────────────────────────┐
│    A          B C D E        F G │
│                                  │
│             H I J K L            │
└──────────────────────────────────┘
2425 . . . . . .34 . . . . . 4647
```

Mit dem CURSOR-Befehl kann man Teile der Anzeige erhalten und an den angegebenen Stellen neue Texte wiedergeben. Der CURSOR-Befehl kann auch für den INPUT-Befehl verwendet werden, wie das folgende Beispiel zeigt:

```
10:WAIT 0
20:PRINT "KOSTEN MENGE  PREIS"
30:CURSOR 24:INPUT A
40:CURSOR 32:INPUT B
50:C=A*B
60:CURSOR 39:PRINT C
70:WAIT : PRINT
80:END
```

DATA

1. DATA ausdruck,ausdruck,ausdruck,ausdruck...

Abkürzung: DA., DAT.
Vergleiche: READ, RESTORE

Wirkung:
Der DATA-Befehl stellt für den READ-Befehl die benötigten Daten bereit.

Anwendung:
Soll ein Feld (Array) mit Daten gefüllt werden, empfiehlt es sich, diese Daten in DATA-Befehlen bereit zu stellen und diese Daten dann mit einer FOR/NEXT-Schleife und einem READ-Befehl in das Feld einzulesen. Der erste READ-Befehl liest den ersten Wert aus der ersten DATA-Zeile, der zweite READ-Befehl den zweiten Wert und so fort. Mit jedem READ-Befehl wird ein Zeiger auf den nächsten Wert gesetzt. Sind die Daten einer DATA-Zeile erschöpft, so wird der Zeiger an den Anfang der nächsten DATA-Zeile gesetzt.

DATA-Befehle haben keinen Einfluß auf den Ablauf des Programms, so daß sie an jeder Stelle des Programms eingefügt werden können. Viele Programmierer setzen DATA-Zeilen gerne hinter die Zeile, in der der dazugehörende READ-Befehl steht. Andere setzen alle DATA-Zeilen an den Anfang oder an das Ende eines Programmes. Wenn nötig, kann der Zeiger mit dem RESTORE-Befehl an den Anfang einer bestimmten DATA-Zeile gesetzt werden. So ist es auch möglich, eine DATA-Zeile mehrmals zu benutzen.

Beispiele:

```
10:DIM B(10)              Initialisierung eines Feldes
20:WAIR 128
30:FOR I=1 TO10           Lädt die Werte aus den DATA-Zeilen in
40:READ B(I)              in das Feld B( ). B(1) wird gleich 10,
50:PRINT B(I)             B(2) gleich 20, B(3) gleich 30 gesetzt,
60:NEXT I                 usw.
70:DATA 10,20,30,40,50,60
80:DATA 70,80,90,100
90:END
```

DEGREE

1. DEGREE

Abkürzung: DE., DEG., DEGR., DEGRE
Vergleiche: GRAD, RADIAN

Wirkung:
Mit dem DEGREE-Befehl werden die Winkelwerte auf Alt-Grad umgeschaltet.

Anwendung:
Der PC 1260/1261 kann drei Formen der Winkeldarstellung verarbeiten:

1. Alt-Grad (dezimal, sexagesimal)
2. Radial-Werte (Bogenmaß)
3. Neu-Grad

Diese Formen werden für die Werte der Argumente von SINus-, COSinus- und TANgens-Funktionen benutzt und als Ergebnisse der Umkehrfunktionen ArcusSINus, ARcusCoSinus und ArcusTaNgens. Mit der DMS- und der DEG-Funktion kann ein Dezimalwinkel in Grad, Minute und Sekunde (sexagesimal) umgeformt werden bzw. umgekehrt.

Beispiel:

```
10:DEGREE
20:X=ASN 1          X hat jetzt den Wert 90, dem Arcussinus von 1.
30:PRINT X
```

DIM

1. DIM dim liste

dim liste	dimensionierung oder dimensionierung,dim liste
dimensionierung	numerische dim oder string dim
numerische dim	numerischer name(größe)
string dim	string name(größe) oder string name(größe)*länge
numerischer name	erlaubte numerische variable
string name	erlaubte string variable
größe	größe oder größe,größe
größe	zahl der elemente
länge	länge eines strings im array

Abkürzung: D., DI.

Wirkung:
Der DIM-Befehl wird benutzt, um Speicher für numerische oder String-Felder frei zu halten.

Anwendung:
Außer für Felder der Form A( ), A$( ) oder Zwei-Zeichen() und Zwei-Zeichen$() muß jedes andere Feld mit einem DIM-Befehl initialisiert werden, um den nötigen Speicherraum bereit zu halten.

Ein Feld kann maximal 2 Dimensionen haben. Die maximale Länge einer Dimension ist 255. Zu der angegebenen Zahl addiert sich ein "nulltes" Element, so daß die Anzahl der Elemente einer Dimension immer um eins größer ist als angegeben, z. B. hat das Feld DIM B(3) die Elemente B(0), B(1), B(2) und B(3). In zweidimensionalen Feldern haben beide Dimensionen ein "Null"-Element.

In String-Feldern kann zusätzlich zur Anzahl der Elemente die Länge der einzelnen Strings vorgegeben werden, z. B. reserviert DIM B$(3)*12 Speicherraum für vier String-Elemente mit je 12 Zeichen Länge. Wird die Länge der Strings nicht vorgegeben, so wird eine maximale Länge von 16 Zeichen angenommen.

Bei der Initialisierung eines Feldes werden alle Werte

a) eines numerischen Feldes gleich Null,
b) eines String-Feldes gleich leer

gesetzt.

Informieren Sie sich bitte für die Initialisierung und Dimensionierung der Felder A() und A$() in dem Abschnitt, in dem die Variablen erklärt werden.

Beispiele:

10:DIM B(10)	Reserviert Speicherraum für ein numerisches Feld mit 11 Elementen.
20:DIM C$(4,4)*10	Reserviert Speicherraum für ein zweidimensionales String-Feld mit 5 Spalten und 5 Reihen. Jeder String soll maximal 10 Zeichen lang sein.

END

1. END

Abkürzung: E., EN.

Wirkung:
Der END-Befehl signalisiert das Ende eines Programms.

Anwendung:
Sind mehrere Programme geladen, so ist es nötig, den einzelnen Programmen Endmarken zu geben, damit bei der Ausführung eines Programms der Rechner nicht in ein anderes hineinläuft. Der Befehl END setzt diese Endmarke.

Beispiel:

```
10:PRINT "HALLO"
20:END
30:PRINT "TSCHUESS"
40:END
```

Wenn Sie das Kommando RUN 10 geben, so gibt der Rechner das Wort HALLO aus, aber nicht das Wort TSCHUESS. Dieses wird auf das Kommando RUN 30 ausgegeben.

FOR... TO...STEP

1. FOR numerische variable=ausdruck1 TO ausdruck2

2. FOR numerische variable=ausdruck1 TO ausdruck2
   STEP ausdruck3

Abkürzung: F., FO. : STE.
Vergleiche: NEXT

Wirkung:
Der FOR-Befehl wird in Verbindung mit dem NEXT-Befehl gegeben, um eine bestimmte Operation mehrmals zu wiederholen.

Anwendung:
Der FOR- und der NEXT-Befehl schliessen eine Gruppe von Befehlen ein, die wiederholt werden sollen. Wird diese Gruppe das erste Mal ausgeführt, so hat die Schleifenvariable den Wert von ausdruck1.

Erreicht das Programm nun den NEXT-Befehl, so wird die Schleifenvariable um den ausdruck3 erhöht und dann mit dem ausdruck2 verglichen. Ist der Wert der Schleifenvariablen kleiner oder gleich dem ausdruck2, so wird die eingeschlossene Gruppe ein weiteres Mal ausgeführt, beginnend mit dem Befehl hinter FOR. In der ersten Form des Befehls ist die Erhöhung (STEP) gleich 1. In der zweiten Form wird die Erhöhung durch den ausdruck3 vorgegeben. Ist der Wert der Schleifenvariablen größer geworden als ausdruck2, so wird das Programm mit dem Befehl hinter NEXT fortgesetzt. Da der Vergleich am Ende der FOR/NEXT-Schleife durchgeführt wird, werden die eingeschlossenen Befehle mindestens einmal ausgeführt.

Der Wert von ausdruck1, ausdruck2 und ausdruck3 muß zwischen -9.999999999 E 99 und 9.999999999 E 99 liegen. Wird ausdruck3 gleich 0 gesetzt, so kann das Programm nicht fortgesetzt werden.

Die Schleifenvariable kann in der umschlossenen Gruppe von Befehlen benutzt werden. Mit ihr kann z. B. ein Feld-Index "hochgezählt" werden. Man sollte aber sehr vorsichtig sein, wenn der Wert der Schleifenvariablen verändert wird.

Programme dürfen niemals so geschrieben werden, daß von einem Befehl außerhalb einer FOR/NEXT-Schleife in eine solche hineingesprungen wird. Ein Programm sollte auch nicht aus einer FOR/NEXT-Schleife heraussspringen. Sie sollten eine FOR/NEXT-Schleife immer über den NEXT-Befehl verlassen. Der Schleifenvariablenwert muß dazu nur größer als ausdruck2 gesetzt werden.

Eine Gruppe von Befehlen, die von einer FOR/NEXT-Schleife eingefasst werden, kann eine weitere FOR/NEXT-Schleife enthalten. Diese muß dann eine andere Schleifenvariable haben und vollständig in der ersten Schleife liegen. Wenn also in einer FOR/NEXT-Schleife ein FOR-Befehl auftaucht, so muß auch der dazugehörenden NEXT-Befehl in der Schleife liegen. Sie können bis zu fünf FOR/NEXT-Schleifen ineinanderlegen.

Beispiele:

```
10:FOR I=1 TO 5
20:PRINT I
30:NEXT I
```

Diese Schleife gibt die Zahlen 1, 2, 3, 4 und 5 aus.

```
40:FOR N=10 TO 0 STEP -1
50:PRINT N
60:NEXT N
```

Diese Schleife zählt rückwärts von 10 nach 0 und gibt die Schleifenvariable aus.

```
70:FOR N= 1 TO 10
80:X=1
90:FOR F=1 TO N
100:X=X*F
110:NEXT F
120:PRINT X
130:NEXT N
```

Diese Schleife berechnet die Fakultäten der Zahlen von 1 bis 10 und gibt das Ergebnis auf der Anzeige wieder.

GOSUB

1. GOSUB ausdruck

Abkürzung: GOS., GOSU.
Vergleiche: GOTO, ON...GOTO, ON...GOSUB, RETURN

Wirkung:
Mit dem GOSUB-Befehl springt das Programm in eine BASIC-Unterroutine.

Anwendung:
Wenn eine Gruppe von Befehlen in einem Programm an verschiedenen Stellen ausgeführt werden soll oder diese Befehlsgruppe in verschiedenen Programmen eingesetzt werden soll, so empfiehlt sich die Verwendung der Befehle GOSUB und RETURN.

Die Befehlsgruppe wird so untergebracht, daß das Programm normalerweise nicht in diese Befehle hineinlaufen kann. Eine Möglichkeit wäre, diese Gruppe hinter den END-Befehl zu stellen. An den Stellen des Hauptprogramms, an der diese Gruppe nun ausgeführt werden soll, fügen Sie den GOSUB-Befehl mit der Zeilennummer der Startzeile der Unterroutine als ausdruck ein. Die letzte Zeile der Unterroutine muß einen RETURN-Befehl enthalten. Wird der GOSUB-Befehl nun ausgeführt, so wird das Programm in der Unterroutine fortgesetzt. Erreicht der PC 1260/1261 den RETURN-Befehl, so springt der Rechner in den Befehl nach dem GOSUB im Hauptprogramm zurück.

In einer Unterroutine kann ein GOSUB vorkommen. Maximal können 10 Unterroutinen ineinandergelegt werden.

Der ausdruck im GOSUB-Befehl darf kein Komma enthalten, z. B. A(2,5) kann nicht benutzt werden. Um von einem bestimmten Punkt mehrere Unterroutinen anspringen zu können, wird der Befehl ON...GOSUB benutzt. Der ausdruck kann nur eine erlaubte Zeilennummer sein, d. h. der Wert muß zwischen 1 und 65279 inklusive liegen oder Sie bekommen die Fehlermeldung ERROR 4.

Beispiel:

```
10:GOSUB 100          Das Programm springt von Zeile 10 in die
20:END                Unterroutine in Zeile 100, gibt das Wort
100:PRINT "HALLO"     HALLO aus und springt dann in die Zeile 20
110:RETURN            des Hauptprogramms zurück.
```

GOTO

1. GOTO ausdruck

Abkürzung: G., GO., GOT.
Vergleiche: GOSUB, ON...GOSUB, ON...GOTO

Wirkung:
Mit dem GOTO-Befehl springt der Rechner an eine bestimmte Stelle eines Programms.

Anwendung:
Der Befehl veranlaßt den PC 1260/1261, von einer bestimmten Stelle in einem Programm an eine andere Stelle durch den ausdruck bestimmte Stelle des Programms zu springen. Anders als im GOSUB-Befehl "erinnert" sich der Rechner nicht, von wo dieser Sprung ausgeführt wurde.

Der ausdruck des Befehls darf kein Komma, wie z. B. in A(1,2) enthalten. Soll von einem Punkt die Möglichkeit gegeben werden, an eine von mehreren Stellen zu springen, so muß der ON...GOTO-Befehl benutzt werden. Der ausdruck muß eine Zeilennummer sein. Erlaubt sind die Werte von 1 bis 65279. Bei einem anderen Wert bekommen Sie die Fehlermeldung ERROR 4.

Gute Programme laufen von Anfang bis Ende ohne Sprünge, abgesehen von Unterroutinensprüngen. Der prinzipielle Gebrauch des GOTO-Befehls ist deshalb im IF...THEN-Befehl zu finden.

Beispiel:

```
10:INPUT A$                    Wenn Sie ein "J" eingeben, gibt der
20:IF A$="J" THEN GOTO 50      Rechner das Wort JA aus, sonst das
30:PRINT "NEIN"                Wort NEIN.
40:GOTO 60
50:PRINT "JA"
60:END
```

GRAD

1. GRAD

Abkürzung: GR., GRA.
Vergleiche: DEGREE, RADIAN

Wirkung:
Mit dem GRAD-Befehl werden die Winkelwerte auf Neu-Grad umgeschaltet.

Anwendung:
Der PC 1260/1261 hat drei Formen der Winkeldarstellung:

1, Alt-Grad
2. Radial-Wert
3. Neu-Grad

Die Argumente der Funktionen SINus, COSinus und TANgens können in dieser Form eingegeben bzw. die Ergebnisse der Umkehrfunktionen ArcusSiNus, ArcusCoSinus und ArcusTaNgens in diesen Formen ausgegeben werden.

Beispiel:

```
10:GRAD
20:X=ASN 1
30:PRINT X
```

X hat jetzt den Wert 100, den Gradienten von Arcussinus 1.

IF...THEN

1. IF bedingung THEN befehl
2. If bedingung befehl

Abkürzung: keine für IF : T., TH., THE

Wirkung:
Das IF...THEN-Befehlspaar wird benutzt, um einen bestimmten Befehl nur dann auszuführen, wenn eine Bedingung erfüllt ist.

Anwendung:
Im normalen Ablauf eines BASIC-Programms werden die Befehle in der Reihenfolge ihres Auftretens abgearbeitet. Das IF...THEN-Befehlspaar erlaubt es, einen Befehl nur dann auszuführen, wenn eine bestimmte Bedingungen erfüllt ist. Ist der Bedingungsteil des IF-Befehls wahr, so wird der Befehl ausgeführt, ist die Bedingung falsch, wird er übersprungen.

Der Bedingungsteil des IF-Befehls kann jeder vergleichende Ausdruck, wie in Kapitel 4 beschrieben, sein. Es ist auch möglich, einen numerischen Ausdruck als Bedingung zu setzen, obgleich der Sinn des Befehls dadurch wenig klar wird. Jeder Ausdruck, der kleiner oder gleich Null ist, setzt die Bedingung falsch. Jeder positive Wert setzt die Bedingung wahr.

Der Befehl, der auf den THEN-Teil des Befehls folgt, kann jeder BASIC-Befehl sein, auch ein weiterer IF...THEN-Befehl. Handelt es sich um eine Zuweisung, so muß der LET-Befehl mit dem Wort LET eingeleitet werden. Ist der auf THEN folgende Befehl kein END, GOTO oder ON...GOTO, so wird der dann folgende Befehl ausgeführt, gleich, ob die Bedingung wahr oder falsch ist.

Die beiden Formen des Befehls sind in der Ausführung gleich, aber die erste Form ist wesentlich klarer.

Beispiel:

```
10:INPUT "WEITER?";A$
20:IF A$="JA" THEN GOTO 10
30:IF A$="NEIN" THEN GOTO 60
40:PRINT "JA ODER NEIN, BITTE"
50:GOTO 10
60:END
```

Dieses Programm wiederholt die Frage WEITER?, solange JA eingegeben wird. Es stoppt, wenn NEIN eingegeben wird. Andere Antworten werden nicht beachtet.

INPUT

1. INPUT eingabe liste

```
eingabe liste    ist eingabe gruppe          oder
                     eingabe gruppe,eingabe liste

eingabe gruppe   ist var list                oder
                     prompt.var liste        oder
                     prompt;var liste

var liste        ist variable                oder
                     variable,var liste

prompt           ist jede String-Konstante

Abkürzung:       I., IN., INP., INPU.
Vergleiche:      INPUT£, READ, CURSOR, PRINT
```

Wirkung:
Der INPUT-Befehl ermöglicht es, einen oder mehrere Werte über die Tastatur einzugeben.

Anwendung:
Wollen Sie in einem Programm die Grundwerte für jeden Programmlauf ändern, so empfiehlt sich der INPUT-Befehl. Die Basiswerte können dann mit der Tastatur eingegeben werden.

In der einfachsten Form fordert der Befehl Sie mit einem Fragezeichen in der linken oberen Ecke der Anzeige auf, die nötige Eingabe zu machen. Sie geben dann den gewünschten Wert, gefolgt von ENTER ein. Dieser Wert wird dann der ersten Variablen der Variablenliste des INPUT-Befehls zugewiesen. Sind in dem Befehl mehrere Variable aufgelistet, so wird der Prozeß so oft wiederholt, bis alle Variablen belegt sind.

Sie können die Eingabe-Aufforderung (=Prompt), also das Fragezeichen, durch einen Prompt-String, das ist eine Bemerkung, die die nötige Eingabe erklären sollte, eigener Wahl ersetzen. Der Ablauf des Zuweisungsprozesses ändert sich nur insofern, daß das Fragezeichen durch diesen String ersetzt wird.

Sollte der Befehl einen Prompt-String und die Variablenliste mehr als eine Variable enthalten, so wird der String nur bei der ersten Aufforderung gezeigt. Die folgenden werden wieder mit dem Fragezeichen gekennzeichnet. Beinhaltet der Befehl einen zweiten Prompt-String, so wird dieser vor der Zuweisung der direkt folgenden Variablen angezeigt.

Wurde vor dem INPUT-Befehl ein CURSOR-Befehl gegeben, so wird das Fragezeichen bzw. der Prompt-String in der durch den CURSOR-Befehl gegebenen Position angezeigt.

Wurde direkt vor dem INPUT-Befehl ein PRINT-Befehl gegeben, der mit einem Semikolon (;) endet, so werden das Fragezeichen bzw. der Prompt-String und die eingegebenen Werte direkt hinter der Ausgabe des PRINT-Befehls angezeigt.

Wenn Sie bei einem INPUT keinen Wert eingeben und direkt die [ENTER] -Taste drücken, so behält die entsprechende Variable ihren alten Wert.

Beispiele:

```
10:INPUT A                    Löscht das Anzeigenfeld und setzt ein
                              Fragezeichen an den Anfang der 1. Zeile.
20:INPUT "A=";A               Gibt "A=" aus und wartet auf eine Eingabe.
30:INPUT "A=",A               Gibt "A=" aus, wartet auf eine Eingabe und
                              zeigt diese dann am Anfang der 1. Zeile an.
                              Der Prompt-String wird also gelöscht.
40:INPUT "X=?";X;"Y=?";Y      Gibt erst das Prompt "X=?" aus, wartet
                              auf eine Eingabe, löscht nach dem [ENTER]
                              die Anzeige und gibt dann den Prompt
                              "Y=?" aus.
```

INPUT=

1. INPUT= var liste
2. INPUT= "dateiname";var liste

var liste ist variable oder
variable,var liste

Abkürzung: I.=, IN.=, INP.=, INPU.=
Vergleiche: INPUT, PRINT=, READ

Wirkung:
Mit dem INPUT=-Befehl können Daten von einer Cassette geladen werden.

Anwendung:
Folgende Variable können im INPUT=-Befehl eingesetzt werden:

1. gesetzte Variable - A, B, C, A(7), D*, A(20)* usw.
2. einfache Variable - AA, B3, CP$ usw.
3. Feld-Variable - S(*), HP(*), K$(*) usw.

1) Übergabe von Daten auf gesetzte Variable

Um Daten von Cassette auf gesetzte Variable zuzuweisen, müssen die Namen der Variablen im Befehl angegeben werden.

```
INPUT= "DATA1";A,B,X,Y
```

Der Befehl weist die Daten der Datei DATA1 den Variablen A, B, X und Y in der gegebenen Reihenfolge zu.

Um alle verfügbaren gesetzten Variablen und, falls definiert, erweiterte Variable (A(27) und weiter) mit Daten von Cassette zu belegen, muß der ersten Variablen ein Stern (*) angehängt werden.

```
INPUT= "D2";D*
```

Mit diesem Befehl werden die Daten der Cassette den Variablen D bis Z und A(27) und höher zugewiesen.

```
INPUT= A(10)* (ohne DIMensionierung)
```

Dieser Befehl überträgt die Daten der ersten gefundenen Datei von Cassette auf die Variablen A(10) und höher (also J bis Z und A(27) und höher).

Achtung:

a) Wurde bereits mit einem DIM-Befehl ein Feld namens "A" definiert, kann keine erweiterte Variable der Form A() definiert werden. Außerdem werden keine Daten auf die Variablen A(27) und höher übertragen.
b) Die Übertragung der Daten auf die gesetzten und erweiterten Variablen (A(27) und höher) wird so lange fortgesetzt, wie Daten gefunden werden, oder bis der Speicher des Rechners voll ist.

2) Datenübertragung auf einfache Variable

Werden im INPUT#-Befehl einfache Variablennamen eingegeben, so werden die Daten der Cassette auf diese Variablen übertragen.

```
INPUT# "DM!";AB,Y1,XY$
```

Dieser Befehl überträgt die Daten von Cassette auf die Variablen AB, Y1 und XY$.

Achtung:

a) Numerische Daten müssen auf numerische Variable, String-Daten auf String-Variable übertragen werden. Andere Zuweisungen sind nicht möglich.

b) Im Programm-Datenspeicherbereich muß Raum für die einfachen Variablen frei gehalten werden, bevor der INPUT#-Befehl ausgeführt werden kann, sonst erhalten Sie eine Fehlermeldung. Die "Platzreservierung" kann mit einer Zuweisung ausgeführt werden.

```
AA=0 ENTER
BI$="A"  ENTER
INPUT# AA;BI$
```

Für die Zuweisung können einfache numerische oder String-Werte benutzt werden.

3) Datenübertragung auf Feldvariable

Um Daten von Cassette auf die Variablen eines Feldes zu übertragen, muß der Name des Feldes in der Form feldname im Befehl angegeben werden.

```
50:DIM B(5)
60:INPUT# "DS4";B(*)
```

Dieser Befehl überträgt die Daten der Datei DS$ auf die Variablen B(0) bis B(5) des Feldes B.

Achtung:

a) Numerische Daten müssen auf numerische Feldvariablen gleicher Länge, String-Daten müssen auf String-Feldvariable gleicher Länge übertragen werden. Wird dies nicht beachtet, erhalten Sie eine Fehlermeldung.

b) Sie müssen Raum im Programm-Datenspeicherbereich bereitstellen, bevor der INPUT#-Befehl ausgeführt wird oder Sie erhalten eine Fehlermeldung. Benutzen Sie dazu den DIM-Befehl.

WICHTIG

Stimmt die Anzahl der Variablen im INPUT#-Befehl nicht mit der Zahl der Daten auf der Cassette überein, geschieht folgendes:

* Ist die Zahl der Daten in der Datei auf Cassette größer als die Zahl der Variablen des Befehls, so wird jeder Variablen ein Wert zugewiesen. Die überzähligen Daten werden ignoriert.

* Ist die Zahl der Daten in der Datei auf Cassette kleiner als die Zahl der Variablen des Befehls, so werden alle Daten der Datei den Variablen zugewiesen und die überzähligen Variablen behalten ihren alten Wert.
  Der Rechner wartet aber auf weitere Daten für diese Variablen. Unterbrechen Sie diesen Status mit der [BRK] -Taste.

* Wird der INPUT#-Befehl ohne Variable eingegeben, so erhalten Sie bei dem Versuch der Ausführung die Fehlermeldung ERROR 1.

LET

1. LET variable=ausdruck
2. variable=ausdruck

Abkürzung: LE.

Wirkung:
Der LET-Befehl weist Variablen Werte zu.

Anwendung:
Mit dem LET-Befehl weisen Sie der angegebenen Variablen den Wert des Ausdrucks zu. Sie können einer numerischen Variablen nur numerische Werte und einer String-Variablen nur String-Werte zuweisen. Um Werte aus der einen in die andere Form zu übertragen, muß eine der beiden Funktionen STR$ oder VAL eingesetzt werden.

Sie können das LET in einer Zuweisung weglassen, es sei denn, die Zuweisung folgt auf das THEN eines IF...THEN-Befehles. Dies ist der einzige Fall, wo das Wort LET im Befehl vorkommen muß.

Beispiele:

```
10:I=10                              Weist I den Wert 10 zu.
20:A=5*I                             Weist A den Wert 50 zu.
30:X$=STR$(A)                        Weist X$ den Wert "50" zu.
40:IF I>=10 THEN LET Y$=X$*".00"     Weist Y$ den Wert "50.00" zu.
```

LPRINT

1. LPRINT druckausdr
2. LPRINT durckausdr,druckausdr,druckausdr,druckausdr
3. LPRINT druckliste
4. LPRINT durckliste;
5. LPRINT

druckliste ist druckausdr oder
druckausdr;druckliste

druckausdr ist ausdruck oder
USING format;ausdruck

Das USING format wird unter USING beschrieben.

Abkürzung: LP., LPR., LPRI., LPRIN.
Vergleiche: PAUSE, PRINT, USING, WAIT

Wirkung:
Mit dem LPRINT-Befehl werden Informationen auf dem optionalen Drucker des Drucker/Mikrocassettenrecorder CE-125 ausgegeben.

Anwendung:
Mit dem LPRINT-Befehl können Sie Ausgaben, die sonst auf der Anzeige des Rechners gemacht werden, mit dem Drucker ausdrucken lassen.

Die erste Form des Befehls druckt einen einzelnen Wert aus. Numerische Ausdrücke werden ganz rechts, String-Ausdrücke ganz links beginnend ausgedruckt.

Die zweite Form teilt die einfache Druckzeile mit 24 Zeichen in zwei Hälften mit 12 Buchstaben. Die eingegebenen Werte werden in der Reihenfolge erster Wert erste Hälfte erste Zeile, zweiter Wert zweite Hälfte erste Zeile, dritter Wert erste Hälfte zweite Zeile, vierter Wert zweite Hälfte zweite Zeile ausgedruckt. Innerhalb der 12-Zeichen-Zeilen-Hälfte werden numerische Ausdrücke wieder ganz rechts und String-Ausdrücke ganz links ausgegeben.

Werte, die mehr als 12 Zeichen lang sind, werden auf 12 Zeichen gekürzt. Numerische werden dabei in Dezimalbruch und Zehnerpotenz ausgegeben. Strings werden nur bis zum 12. Zeichen ausgedruckt.

Achtung: Form 2 muß 2-4 Ausdrücke haben, sonst gibt es eine Fehlermeldung. Selbst, wenn die Wiedergabe-Startposition für die Form 2 durch Form 4 wiedergegeben ist, wird diese gelöscht und der Druck erfolgt wie oben beschrieben.

Form 3 druckt die Werte links beginnend aus. Werden mehr als 24 Zeichen eingegeben, so wird der Druck in der nächsten Zeile fortgesetzt. Maximal können 79 Zeichen ausgedruckt werden. Liegt die 79. Spalte in der Mitte eines numerischen Wertes, so erhalten Sie eine Fehlermeldung.

Form 4 des Befehls setzt mit dem Semikolon den Startpunkt des nächsten LPRINT-Befehls fest. Dieser beginnt direkt hinter dem zuletzt ausgegebenen Wert.

Die Form 5 des Befehls druckt nichts aus. Sie führt nur einen Zeilenvorschub aus.

Beispiele:

```
10:A=10;B=20;X$="ABCD";Y$="XYZ"
20:LPRINT A
30:LPRINT X$
40:LPRINT A,B,X$,Y$
50:LPRINT X$;A;B
60:LPRINT
70:LPRINT A*Z;
80:LPRINT Y$
```

NEXT

1. NEXT numerische variable

Abkürzung: N., NE., NEX.
Vergleiche: FOR

Wirkung:
Der NEXT-Befehl kennzeichnet das Ende einer Gruppe von Befehlen, die in einer FOR/NEXT-Schleife wiederholt werden sollen.

Anwendung:
Der NEXT-Befehl tritt immer in Verbindung mit einem FOR-Befehl auf und wird deshalb dort ausführlich erklärt. Die Variable des NEXT muß mit der des dazugehörigen FOR übereinstimmen.

Beispiel:

```
10:FOR I=1 TO 10
20:PRINT I
30:NEXT I
```

ON...GOSUB

1. ON ausdruck GOSUB ausdr liste

ausdr liste ist ausdruck oder
ausdruck,ausdr liste

Abkürzung: O. : GOS., GOSU.
Vergleiche: GOSUB, GOTO, ON...GOTO

Wirkung:
Der ON...GOSUB-Befehl wird benutzt, um eine der gegebenen Unterroutinen abhängig von einem Wert auszuführen.

Anwendung:
Bei der Ausführung des ON...GOSUB-Befehls wird der ausdruck zwischen ON und GOSUB auf einen ganzzahligen Wert reduziert. Ist der Wert dieses Ausdrucks 1, so wird die erste der durch ausdr liste angegebenen Unterroutinen nach dem GOSUB ausgeführt. Ist der Wert 2, so wird die zweite ausgeführt usw. Nach dem RETURN aus der Unterroutine wird der Befehl der direkt auf den ON...GOSUB-Befehl folgt, ausgeführt.

Wenn der Wert des Ausdrucks kleiner 1 oder größer der Zahl der gegebenen Unterroutinen ist, wird das Programm mit dem auf den ON...GOSUB-Befehl folgenden Befehl fortgesetzt.

Achtung: In den Ausdrücken nach dem GOSUB dürfen keine Kommata vorkommen. Der PC 1260/1261 kann nicht Kommata in Ausdrücken von Kommata zwischen Ausdrücken unterscheiden.

Beispiel:

```
10:INPUT A
20:ON A GOSUB 100,200,300
30:END
100:PRINT "ERSTE"
110:RETURN
200:PRINT "ZWEITE"
210:RETURN
300:PRINT "DRITTE"
310:RETURN
```

Wird in Zeile 10 "1" eingegeben, so gibt der Rechner "ERSTE" aus. Auf "2" gibt er "ZWEITE" und auf "3" "DRITTE" aus. Auf andere Eingaben wird nichts ausgegeben.

ON...GOTO

1. ON ausdruck GOTO ausdr liste

ausdr liste ist ausdruck oder
ausdruck,ausdr liste

Abkürzung: O.; G., GO., GOT.
Vergleiche: GOTO, GOSUB, ON...GOSUB

Wirkung:
Mit dem Befehl ON...GOTO wird es möglich, das Programm abhängig von einem Wert an einer bestimmten Stelle fortzusetzen.

Anwendung:
Bei der Ausführung des Befehls ON...GOTO wird der ausdruck zwischen ON und GOTO auf einen ganzzahligen Wert reduziert. Ist der Wert dieser Zahl 1, so wird das Programm an der ersten angegebenen Stelle fortgesetzt. Ist er 2, an der zweiten usw.

Ist der Wert des Ausdrucks kleiner 1 oder größer der Zahl der gegebenen Sprungziele, so wird der auf den ON...GOTO-Befehl folgende Befehl ausgeführt.

Achtung: In den Ausdrücken, die auf das GOTO folgen, dürfen keine Kommata vorkommen. Der PC 1260/1261 ist nicht in der Lage, Kommata, die in Ausdrücken auftreten, von Kommata, die zwischen Ausdrücken stehen, zu unterscheiden.

Beispiel:

```
10:INPUT A                  Die Eingabe '1' läßt den Rechner
20:ON A GOTO 100,200,300    'ERSTE', '2' 'ZWEITE' und '3'
30:GOTO 900                 'DRITTE' ausgeben.
100:PRINT "ERSTE"
110:GOTO 900
200:PRINT "ZWEITE"
210:GOTO 900
300:PRINT "DRITTE"
900:END
```

PAUSE

1. PAUSE druck ausdr
2. PAUSE druck ausdr,druck ausdr,druck ausdr,druck ausdr
3. PAUSE druck liste
4. PAUSE druck liste;
5. PAUSE

druck liste ist druck ausdr oder
druck ausdr;druck liste

druck ausdr ist ausdruck oder
USING format;ausdruck

Das USING format wird unter USING beschrieben.

Abkürzung: PAU., PAUS.
Vergleiche: LPRINT, PRINT, CURSOR, USING, WAIT

Wirkung:
Mit dem PAUSE-Befehl wird eine Ausgabe auf dem Display zeitlich begrenzt wiedergegeben.

Anwendung:
Der PAUSE-Befehl wird benutzt, um kurze Texte, Ergebnisse usw. anzuzeigen. Die Ausführung des Befehls ist die gleiche wie die des PRINT-Befehls. Mit einer Ausnahme: Nach der Ausgabe des Textes oder des Wertes macht der PC 1260/1261 eine Pause von ca. 0,85 Sekunden, bevor der nächste Befehl ausgeführt wird. Ein WAIT-Intervall oder [ENTER] wird nicht gebraucht.

Die erste Form des PAUSE-Befehls gibt einen einzelnen Wert aus. Numerische Werte werden ganz rechts, String-Werte ganz links beginnend ausgegeben. Wurde die Startposition von einem USING-Befehl (Form 4) oder einem CURSOR-Befehl vorgegeben, beginnt die Ausgabe an dieser Stelle.

In Form 2 wird das Anzeigenfeld in vier Gruppen mit je 12 Zeichen unterteilt. Der erste angegebene Wert wird in der oberen Zeile, linke Hälfte, der zweite in der oberen Zeile, rechte Hälfte, der dritte in der unteren Zeile, linke Hälfte und der vierte in der unteren Zeile, rechte Hälfte wiedergegeben. Auch hier werden innerhalb des Blocks numerische Werte ganz links und String-Werte ganz rechts beginnend ausgegeben.

Selbst, wenn die Startposition durch einen vorhergehenden PAUSE-Befehl der Form 4 oder durch einen CURSOR-Befehl vorgegeben wurde, werden die Werte in der Form 2 wie oben wiedergegeben.

* Die Anzahl der Werte in Form 2 muß zwischen 2 und 4 liegen.
* Ist ein Wert mehr als 12 Spalten (Zeichen) lang, wird
a) ein numerischer Wert als Dezimalbruch mit Exponent wiedergegeben,
b) ein Stringwert auf seine ersten 12 Zeichen gekürzt.

Form 3 gibt die Werte in direkter Folge, beginnend in der oberen Zeile, links wieder. Die Startposition kann mit einem CURSOR- oder USING-Befehl vorgegeben werden.

Achtung: Wenn der Wert mehr Zeichen hat als die Anzeigefelder, so werden die überzähligen ignoriert. Ist die Zeichenanzahl größer 79, erhalten Sie eine Fehlermeldung, wenn die 79. Spalte in einem numerischen Wert liegt.

Form 4 des Befehls gibt die eingegebene Liste entsprechend in der oberen Zeile links wieder. Die auf das letzte Zeichen folgende Spalte ist die Startposition des nächsten Ausgabebefehls (z. B. PRINT):

Form 5 gibt den vorangegangenen Wert unverändert wieder.

Beispiele:

```
10:A=10:B=20:X$="ABCDEF":Y$="XYZ"
```

	ANZEIGE
`20:PAUSE A`	10.
`30:PAUSE X$`	ABCDEF
`40:PAUSE X$,Y$,A,B`	ABCDEF XYZ 10. 20.
`50:PAUSE Y$;X$;`	XYZABCDEF
`60:PAUSE A*B`	XYZABCDEF200

PRINT

1. PRINT druck ausdr
2. PRINT druck ausdr,druck ausdr,druck ausdr,druck ausdr
3. PRINT druck liste
4. PRINT druck liste;
5. PRINT PRINT
6. PRINT = LPRINT
7. PRINT = PRINT

druck liste ist druck ausdr oder
druck ausdr;druck liste

druck ausdr ist ausdruck oder
USING format;ausdruck

Das USING format wird bei USING beschrieben.

Abkürzung: P., PR., PRI., PRIN.
Vergleiche: LPRINT, PAUSE, CURSOR, USING, WAIT

Wirkung:
Mit dem PRINT-Befehl können Ausgaben auf der Anzeige des PC 1260/1261 oder dem Drucker des Drucker/Mikrocassettenrecorders CE-125 gemacht werden.

Anwendung:
Mit dem PRINT-Befehl werden Texte und Ergebnisse ausgegeben. Die erste Form des Befehls gibt einen einzelnen Wert aus. Numerische Werte werden ganz rechts, String-Werte ganz links beginnend wiedergegeben. Die Startposition kann mit einem CURSOR- oder einem USING-Befehl (wie in Form 4) vorgegeben werden.

Die Form 2 des Befehls teilt die Anzeige in vier Gruppen zu je 12 Zeichen. Der erste Wert wird in der oberen Zeile, linke Hälfte, der zweite in der oberen Zeile, rechte Hälfte, der dritte in der unteren Zeile linke Hälfte und der vierte Wert in der unteren Zeile, rechte Hälfte, wiedergegeben. Die Startposition der Werte in den Feldern ist, unabhängig von irgendwelchen Vorgaben, für numerische Werte ganz rechts, für String-Werte ganz links.

* Die Anzahl der Werte in Form 2 muß zwischen 2 und 4 liegen.
* Ist ein Wert länger als 12 Spalten (Zeichen), so wird
a) ein numerischer Wert als Dezimalbruch mit Exponent wiedergegeben,
b) ein String-Wert auf die ersten 12 Zeichen gekürzt ausgegeben.

In Form 3 des Befehls werden die Werte direkt hintereinander, beginnend in der oberen Zeile links, wiedergegeben. Die Startposition kann mit einem USING- (wie in Form 4) oder einem CURSOR-Befehl vorgegeben werden.

Achtung: Sollen mehr als 48 Zeichen ausgegeben werden, so werden die überzähligen Zeichen ignoriert. Sie erhalten eine Fehlermeldung (ERROR 6), wenn die Anzahl der zeichen 79 Zeichen überschreitet und die 79. Spalte in einem numerischen Wert liegt.

Form 4 beginnt die Ausgabe in der oberen Zeile. Die Startposition für einen folgenden Ausgabefehl (z. B. PRINT) wird durch Form 4 des PRINT-Befehls auf die Spalte nach dem letzten Zeichen des letzten Wertes vorgegeben.

Form 5 gibt den vorangegangenen PRINT-Wert unverändert wieder.

Die Formen 6 und 7 geben nichts aus. Die 6. Form setzt den PRINT-Befehl gleich dem LPRINT-Befehl. Alle Ausgaben werden jetzt also auf dem Drucker des CE-125 und nicht mehr auf der Anzeige des PC 1260/1261 gemacht. Mit der 7. Form wird die 6. Form wieder zurückgesetzt, die Ausgabe erfolgt wieder über die Anzeige des PC 1260/1261.

Es ist möglich, in einem PRINT-Befehl mehr als 48 Zeichen anzugeben. Mehr als 48 Zeichen können auf der Anzeige des Rechners aber nicht wiedergegeben werden. Die folgenden Zeichen (49. und höher) können nicht mehr über die Anzeige ausgegeben werden.

Beispiele:

```
10:A=123:B=5/):X$="ABCDEF":Y$="VWXYZ"
```

	ANZEIGE
20:PRINT X$,B	ABCDEF 5.55555E-01
30:PRINT A;B	123.5.5555555556E-01
40:PRINT X$;A	ABCDEF123
50:PRINT Y$;B	ABCDEF123.VWXYZ5.5555555 56E-01

PRINT=

1. PRINT= var liste
2. PRINT= "dateiname";var liste

var liste ist variable oder
variable,var liste

Abkürzung: P.=, PR.=, PRI.=, PRIN.=
Vergleiche: INPUT=, PRINT, READ

Wirkung:
Mit dem PRINT=-Befehl werden Werte auf Cassette abgespeichert.

Anwendung:
Folgende Variablentypen können benutzt werden:

1. gesetzte Variable - A, B, X, A(26), C*, A(10)* usw.
2. einfache Variable - AA, B2, XY$ usw.
3. Feld-Variable - B(*), CD(*), N$(*) usw.

1) Abspeichern von gesetzten Variablen

Um die Werte von gesetzten Variablen auf Cassette abzuspeichern, müssen die Namen der Variablen im PRINT=-Befehl angegeben werden.

```
PRINT= "DATEI1";A,B,X,Y
```

Dieser Befehl speichert die Inhalte der Variablen A, B, X und Y in der Datei "DATEI1" auf Cassette ab.

Wenn Sie den Inhalt aller Variablen ab einer bestimmten abspeichern wollen, so geben Sie die Startvariable gefolgt von einem Stern (*) ein.

```
PRINT= "D2";D*
```

Dieser Befehl speichert die Inhalte der Variablen D bis Z (und A(26) und höher, wenn definiert) in der Datei mit dem Dateinamen "D" ab.

```
PRINT=E,X$,A(30)*
```

Dieser Befehl speichert die Inhalte der gesetzten Variablen E und X$ sowie die Inhalte der erweiterten Variablen A(30) und höher auf Cassette ab.

Achtung: A() kann nicht zur Definition nachfolgender Variablen benutzt werden, wenn A bereits durch einen DIM-Befehl definiert wurde. Ansonsten können die Variablen A(1) bis A(26) im PRINT=-Befehl genauso angegeben werden, wie die Variablen A bis Z bzw. A$ bis Z$.

2) Abspeicherung der Inhalte von einfachen Variablen

Die Inhalte von einfachen Variablen (Zwei-Zeichen-Variablen) werden abgespeichert, indem die Namen der Variablen im PRINT=-Befehl angegeben werden.

```
PRINT= "DM1";AB,Y1,XY$
```

Dieser Befehl speichert die Inhalte der einfachen Variablen AB, Y! und XY$ in der Datei mit dem Dateinamen "DM1" ab.

3) Abspeicherung der Inhalte von Feldvariablen

Die Inhalte aller Variablen eines Feldes können auf Cassette abgespeichert werden, indem der Name des Feldes im Befehl angegeben und die Variablen selbst mit einem Stern (*) umschrieben werden.

```
PRINT= "DS2";X(*),Y(*)
```

Dieser Befehl speichert die Inhalte der Elemente der Felder X (X(0),X(1),...) und Y (Y(0),Y(1)...) in der Datei mit dem Dateinamen "DS2" ab.

Achtung: Es ist nicht möglich, den Inhalt eines oder einiger Elemente eines Feldes gesondert abzuspeichern.

Während gesetzte und erweiterte Variable zulassen, daß nur ein spezifizierter Teil von ihnen auf Cassette gespeichert wird, ist dies mit einem durch den DIM-Befehl definierten Feld nicht möglich.

* Wird der PRINT=-Befehl ohne Variable eingegeben, erhalten Sie die Fehlermeldung ERROR 1.

WICHTIG

Die Lokation von erweiterten (A(26) und höher), einfachen und/oder Feldvariablen muß vor Ausführung des PRINT=-Befehls im Programm/Datenspeicherbereich bereitgestellt werden. Sonst ist die Ausführung des Befehls nicht möglich und Sie erhalten eine Fehlermeldung.

RADIAN

1. RADIAN

Abkürzung: RAD., RADI., RADIA.
Vergleiche: DEGREE, GRAD

Wirkung:
Der RADIAN-Befehl wechselt die Darstellung der Winkelwerte in die Radialform (Bogenmaß).

Anwendung:
Der PC 1260/1261 hat drei Möglichkeiten der Darstellung von Winkelwerten:

- Alt-Grad
- Radial-Wert
- Neu-Grad

Sie werden für die Argumente der Funktionen SINus, COSinus und TANgens sowie für die Ergebnisse der Umkehrfunktionen ArcusSiNus, ArcusCoSinus und ArcusTaNgens gebraucht.

Der RADIAN-Befehl wechselt die Darstellungsform auf Radialwert. Die Radialform gibt den Winkel als Bogenmaß A in Abhängigkeit vom Radius wieder. 360 Grad sind z. B. 2 PI, da der Einheitskreis den Umfang 2 mal PI mal Radius hat.

Beispiel:

```
10:RADIAN
20:X=ASN 1          X hat den Wert 1.570796327 (=PI/2), Arcsin von 1.
30:PRINT  X
```

RANDOM

1. RANDOM

Abkürzung. RA., RAN., RAND., RANDO.

Wirkung:
Der RANDOM-Befehl setzt eine neue Startzahl für den Zufallszahlengenerator.

Anwendung:
Wenn mit der RND-Funktion Zufallszahlen erzeugt werden, so startet der PC 1260/1261 bei einer vorgegebenen Zahl. Der RANDOM-Befehl setzt die Startzahl auf einen neuen, zufällig gewählten Wert.

Die Startzahl für den Zufallszahlengenerator ist jedes Mal, wenn der PC 1260/1261 eingeschaltet wird, die gleiche. Auch die Sequenz der Zufallszahlen wiederholt sich damit, es sei denn, die Basis(oder Start-)zahl wird mit dem RANDOM-Befehl gewechselt. Diese Eigenschaft ist sehr wichtig, ist es doch möglich, den korrekten Ablauf eines Programms zu ermitteln, da auch dann, wenn im Programm RND-Befehle auftreten, das Verhalten des Rechners immer gleich sein muß. Mit dem RANDOM-Befehl verbessern Sie also das Zufallsverhalten Ihres Programms, da die Basiszahl des Zufallsgenerators auch zufällig ermittelt wird.

Beispiel:

```
10:RANDOM
20:X=RND10
30:GOTO 20
```

Bei Start in Zeile 20 wird die Standardbasis des Zufallszahlengenerators eingesetzt. Bei Start in Zeile 10 wird eine neue Basis ermittelt.

READ

1. READ var liste

var liste ist variable oder
variable,var liste

Abkürzung: REA.
Vergleiche: DATA, RESTORE

Wirkung:
Der READ-Befehl wird gebraucht, um Daten aus den DATA-Zeilen herauszulesen und Variablen zuzuweisen.

Anwendung:
Wenn einem Feld Basiswerte zugewiesen werden sollen, so ist es sinnvoll, diese Daten in DATA-Zeilen unterzubringen und sie von dort mit einem READ-Befehl in einer FOR/NEXT-Schleife in das Feld zu übertragen. Wird der erste READ-Befehl ausgeführt, so wird der erste Wert aus den DATA-Zeilen zugewiesen, mit dem zweiten READ-Befehl der zweite Wert usw., unabhängig davon, wieviele DATA-Zeilen oder Werte in den DATA-Zeilen vorhanden sind.

Wenn nötig, können dieselben Daten ein zweites Mal gelesen werden. Hierzu wird der RESTORE-Befehl gebraucht.

Beispiel:

```
10:DIM B(10)          Dimensionierung eines Feldes
20:WAIT 32
30:FOR I=1 TO 10      Zuweisung der Werte aus den DATA-Zeilen
40:READ B(I)          in die Feldvariablen B(0) bis B(10);
50:PRINT B(1)*2       B(0) ist 10, B(1) ist 20 usw.
60:NEXT I
70:DATA 10,20,30,40,50,60
80:DATA 70,80,90,100
90:END
```

REM

1. REM kommentar

Wirkung:
Mit dem REM-Befehl können Kommentare in ein Programm eingefügt werden.

Anwendung:
Es ist sinnvoll, in ein Programm erläuternde Kommentare einzufügen. Es kann sich hierbei um Titel, Autorennamen, Daten der letzten Änderung, Anwendungshinweise uvm. handeln.

Solche Kommentare können mit einem REM-Befehl eingefügt werden. Der REM-Befehl hat keinen Einfluß auf den Programmablauf. Er kann überall im Programm eingefügt werden. Alles, was hinter einem REM-Befehl steht, wird als Kommentar behandelt. Deshalb muß der Befehl REM immer der letzte in einer Zeile sein.

Beispiel:

```
10:REM DIESE ZEILE HAT KEINEN EFFEKT
```

RESTORE

1. RESTORE
2. RESTORE ausdruck

Abkürzung: RES., REST., RESTO., RESTR.
Vergleiche: DATA, READ

Wirkung:
Der RESTORE-Befehl ermöglicht es, eine DATA-Zeile mehrmals zu lesen oder die Reihenfolge, in der die DATA-Zeilen gelesen werden sollen, zu verändern.

Anwendung:
Normalerweise werden mit dem READ-Befehl die Daten aus den DATA-Zeilen eine nach dem anderen herausgelesen, in der Reihenfolge, in der sie stehen. Ist der letzte Wert aus einer DATA-Zeile herausgelesen, so wird, wenn vorhanden und abgefragt, der erste Wert aus der nächsten DATA-Zeile genommen.

Die erste Form des RESTORE-Befehls setzt den Zeiger an den Anfang der ersten DATA-Zeile zurück.

Die zweite Form setzt den Zeiger an den Anfang der ersten DATA-Zeile, deren Zeilennummer größer oder gleich der durch ausdruck angegebenen Zeilennummer.

Beispiel:

```
10:DIM B(10)          Dimensionierung eines Feldes
20:WAIT 32
30:FOR I=1 TO 10      Setzt den Zeiger an den Anfang der 1. DATA-Zeile.
40:RESTORE            Den Variablen von B werden die Werte zugewiesen.
50:READ B(1)          Alle Elemente haben den Wert 20.
60:PRINT B(1)*I;
70:NEXT I
80:DATA 20
90:END
```

RETURN

1. RETURN

Abkürzung: RE., RET., RETU., RETUR.
Vergleiche: GOSUB, ON...GOSUB

Wirkung:
Mit dem RETURN-Befehl wird eine Unterroutine geschlossen, der Programmablauf wird mit dem Befehl, der auf den GOSUB-Befehl folgt, fortgesetzt.

Anwendung:
Eine Unterroutine kann mehr als ein RETURN enthalten. Erreicht das Programm einen dieser RETURN-Befehle, wird die Unterroutine geschlossen und das Programm mit dem Befehl hinter dem GOSUB bzw. ON...GOSUB, von dem der Rechner in die Unterroutine gesprungen ist, fortgesetzt. Wird ein RETURN ohne GOSUB ausgeführt, erhalten Sie die Fehlermeldung ERROR 5.

Beispiel:

```
10:GOSUB 100          Das Programm druckt "HALLO" aus und springt
20:END                zurück in Zeile 20.
100:PRINT "HALLO"
110:RETURN
```

STOP

1. STOP

Abkürzung: S., ST., STO.
Vergleiche: END; CONT-Kommando

Wirkung:
Mit dem STOP-Befehl wird die Ausführung eines Programms gestoppt.

Anwendung:
Erreicht der PC 1260/1261 in einem Programm einen STOP-Befehl, so wird die Ausführung des Programms abgebrochen und Sie erhalten eine Meldung wie z. B. "BREAK in 200", wenn das Programm in Zeile 200 gestoppt wurde. Sie können jetzt die Variablen des Programms überprüfen und so die Ausführung desselben kontrollieren. Die Ausführung des Programms kann mit dem CONT-Befehl fortgesetzt werden.

Beispiel:

```
10:STOP
```

BREAK IN 10 erscheint auf der Anzeige.

TROFF

1. TROFF

Abkürzung: TROF.
Vergleiche: TRON

Wirkung:
Mit dem TROFF-Befehl wird der Trace-Modus ausgeschaltet.

Anwendung:
Die Ausführung des Befehls TROFF hat zur Folge, daß ein Programm wieder auf "normale" Weise ausgeführt wird.

Beispiel:

```
10:TRON
20:FOR I=1 TO 3
30:NEXT I
40:TROFF
```

Bei Ausführung des Programms werden die Zeilennummern ausgegeben:
10, 20, 30, 30 und 40

TRON

1. TRON

Abkürzung: TR., TRO.
Vergleiche: TROFF

Wirkung:
Der TRON-Befehl schaltet den Trace-Modus ein.

Anwendung:
Der Trace-Modus unterstützt die Fehlersuche in Programmen. Ist der Trace-Modus eingeschaltet, so wird die Zeilennummer der gerade ausgeführten Zeile auf der Anzeige ausgegeben. Der PC 1260/1261 wartet dann auf den Runter-Pfeil, um die nächste Zeile auszuführen. Mit dem Hoch-Pfeil kann die gerade ausgeführte Zeile zur Anzeige gebracht werden. Der Trace-Modus bleibt so lange eingeschaltet, bis er mit dem TROFF-Befehl wieder ausgeschaltet wird.

Beispiel:

```
10:TRON
20:FOR I=1 TO 3
30:NEXT I
40:TROFF
```

Bei Ausführung des Programms werden die Zeilennummern ausgegeben:
10, 20, 30, 30 und 40

USING

1. USING
2. USING "spezifikation"
3. USING zeichen variable

Abkürzung: U., US., USI., USIN.
Vergleiche: LPRINT, PAUSE, PRINT

Weitere Erklärung zu USING finden Sie im Anhang C.

Wirkung:
Mit dem USING-Befehl kann das Anzeigenfeld oder der Druck formatiert werden.

Anwendung:
Der USING-Befehl kann einzeln oder als Erweiterung der Befehle LPRINT, PAUSE oder PRINT eingesetzt werden. Der USING-Befehl erstellt eine Ausgabeformatierung, die für alle folgenden Ausgabefehle Gültigkeit hat, bis die Formatierung durch einen neuen USING-Befehl geändert wird.

Die Spezifikationen des Befehls bestehen aus einer in Anführungsstrichen (") gesetzten Zeichenkette. Die Kette ist aus folgenden Zeichen zusammenzusetzen:

= Rechtsbündiges Feld für numerische Zeichen

. Dezimalpunkt

^ Zahlen sollen in Exponentialschreibweise wiedergegeben werden.

& Linksbündiges Feld für alphanumerische Zeichen

Die Spezifikation "====" erstellt z. B. ein rechtsbündiges Feld für numerische Zeichen mit Platz für drei Ziffern und das Vorzeichen. In numerischen Feldern muß immer eine Position für das Vorzeichen vorgesehen sein, auch, wenn nur positive Zahlen ausgegeben werden sollen.

Spezifikationen können mehr als ein Feld beschreiben. "====&&&&" setzt z. B. ein numerisches Feld und ein alphanumerisches Feld direkt nebeneinander.

Wird, wie in Form 1 des Befehls, keine Spezifikation gegeben, so werden alle gesetzten Formate gelöscht und die Standardformatierung tritt wieder in kraft.

Beispiele:

```
10:A=125:X$="ABCDEF"
20:PRINT USING "==.==";A
30:PRINT USING "&&&&&&&&";X$
40:PRINT USING "====&&&";A;X$
```

ANZEIGE

```
 1.25 E 02
ABCDEF
125ABC
```

WAIT

1. WAIT
2. WAIT ausdruck

Abkürzung: W., WA., WAI.
Vergleiche: PAUSE, PRINT

Wirkung:
Der WAIT-Befehl setzt die Zeit fest, die eine Ausgabe auf der Anzeige des PC 1260/1261 wiedergegeben werden soll.

Anwendung:
Bei normaler Ausführung eines Programms wartet der PC 1260/1261 nach einem PRINT-Befehl, bis die [ENTER] -Taste gedrückt wird. Der WAIT-Befehl weist den PC 1260/1261 nun an, eine Ausgabe für eine vorgegebene Zeit anzuzeigen und dann die Ausführung des Programms fortzusetzen, genau wie mit dem PAUSE-Befehl. Der ausdruck im WAIT-Befehl setzt die Länge des Intervalls fest. Er kann jeden Wert von 0 bis 65535 haben. Dabei bedeutet die Erhöhung um 1 eine Verlängerung des Intervalls um 1/59 Sekunde. WAIT 0 ist zu schnell, als daß die Ausgabe gelesen werden könnte; WAIT 65535 setzt das Intervall auf ca. 19 Minuten. WAIT ohne Ausdruck setzt den PC 1260/1261 auf seine normale Arbeitsweise zurück.

Beispiel:

```
10:WAIT 59          Der Rechner wartet nach PRINT etwa 1 Sekunde.
```

FUNKTIONEN

Pseudovariable sind eine Gruppe von Funktionen, die ohne Argument wie einfache Variable behandelt werden.

INKEY$

1. INKEY$

INKEY$ ist eine String-Pseudovariable, die den Wert der letzten, auf der Tastatur gedrückten Taste zugewiesen bekommt. Die Tasten [ENTER], [CL], [SHIFT], [DEF], [↑]-Pfeil, [↓]-Pfeil, [►]-Pfeil und [◄]-Pfeil haben den Wert NUL (=leer). Mit INKEY$ können Eingaben über die Tastatur ohne ein abschließendes [ENTER] gemacht werden.

```
10:A$= INKEY$
20:B=ASC A$
30:IF B=0 THEN GOTO 10
40:IF B=...
```

Die Zeilen ab Zeile 40 können eine Tastatur-Testroutine enthalten. Wenn das Programm gestartet wird, ist der Wert von INKEY$=NUL, da die letzte gedrückte Taste [ENTER] war. Folgt INKEY$ auf einen PRINT- oder PAUSE-Befehl, so nimmt INKEY$ den Wert der Anzeige an.

MEM

1. MEM

MEM ist eine numerische Pseudovariable, die den Wert der Anzahl der im Speicher verbliebenen Zeichen zugewiesen bekommt. Der verfügbare Speicher ist die Differenz zwischen Gesamtspeicherkapazität und mit Programmen und Feld-Variablen belegtem Speicher. MEM kann auch als Kommando benutzt werden.

Im PC 1260 hat MEM nach eine Neuinitialisierung (RESET) den Wert 3198, im PC 1261 den Wert 9342, d. h. im PC 1260 sind 3198 Bytes frei, im PC 1261 sind 9342 Bytes frei.

PI

1. PI

PI ist eine numerische Pseudovariable, die den Wert π hat. Sie hat die gleiche Funktion wie die spezielle Taste [π] auf der Tastatur. Wie alle anderen Zahlen wird auch der Wert von PI mit einer Genauigkeit von 10 Stellen wiedergegeben (3.141592654).

NUMERISCHE FUNKTIONEN

Numerische Funktionen sind eine Gruppe von mathematischen Funktionen, die aus einem numerischen Argument ein numerisches Ergebnis errechnen. Die Gruppe enthält trigonometrische und logarithmische Funktionen, sowie solche, die ganze Zahlen und Beträge errechnen bzw. Vorzeichen ermitteln. Viele BAISC-Dialekte fordern, daß das Argument der Funktion in Klammern steht. Der PC 1260/1261 braucht Klammern nur dann, wenn in einer komplexen Berechnung das Argument der Funktion genau abgegrenzt werden muß.

```
LOG 100+100     wird als

(LOG 100)+100   und nicht als     LOG (100+100)
```

interpretiert.

ABS

1. ABS numerischer ausdruck

Errechnet den Absolut-Betrag des angegebenen Arguments.

```
ABS -10 ist gleich 10
```

ACS

1. ACS numerischer ausdruck

Errechnet den Arcuscosinus des Arguments. Arcuscosinus ist die Umkehrfunktion des Cosinus. Das Ergebnis ist also der Winkel zu dem angegebenen Wert. Die Darstellung des Ergebnisses ist abhängig vom Winkelmodus des PC 1260/1261 (DEG, RAD, GRAD).

```
ACS .5     ist im DEG-Modus gleich 60
```

ASN

1. ASN numerischer ausdruck

Errechnet den Arcussinus des angegebenen Arguments. Arcussinus ist die Umkehrfunktion des Sinus. Die Darstellung des Ergebnisses ist abhängig vom Winkelmodus des PC 1260/1261 (DEG, RAD, GRAD).

```
ACS .5     ist im DEG-Modus gleich 30
```

ATN

1. ATN numerischer Ausdruck

Das Ergebnis ist der Arcustangens des Arguments. Arcustangens ist die Umkehrfunktion des Tangens. Die Darstellung des Ergebnisses ist abhängig vom Winkelmodus des PC 1260/1261 (DEG, RAD, GRAD).

ATN 1 ist im DEG-Modus gleich 45.

COS

1. COS numerischer ausdruck

Errechnet den Cosinus eines Winkels. Die Darstellung des Ergebnisses ist abhängig vom Winkelmodus des PC 1260/1261 (DEG, RAD, GRAD).

COS 60 ist im DEG-Modus gleich .5

DEG

1. DEG numerischer ausdruck

Das Ergebnis ist die Umwandlung des Winkelarguments (Alt-Grad) als DMS-Darstellung (Grad, Minute, Sekunde) in die Dezimaldarstellung DEG. Im DMS-Format stellt der ganzzahlige Anteil der Zahl die Gradzahl, die ersten beiden Dezimalstellen die Minuten und die dritte und vierte Dezimalstelle die Sekunden dar. Die weiteren Stellen geben die Dezimalsekunden wieder. 55 Grad 10' 44,5" werden also z. B. als 55.10445 dargestellt. Im DEG-Format ist der ganzzahlige Anteil wieder Grad, aber die Dezimalstellen sind Dezimalgradbruchteile.

DEG 55.10445 ist also gleich 55.17902778.

DMS

1. DMS numerischer ausdruck

DMS ist die Umkehrfunktion zur DEG-Funktion (vergleiche DEG).

DMS 55.17902778 ist gleich 55.10445.

EXP

1. EXP numerischer ausdruck

Errechnet die Potenzen der Zahl e (=2.718181828, die Basis des Logarithmus naturalis).

EXP 1 ist gleich 2.718281828

INT

1. INT numerischer ausdruck

Das Ergebnis ist der ganzzahlige Anteil des Arguments.

INT PI ist gleich 3.

LOG

1. LOG numerischer ausdruck

Errechnet den Logarithmus zur Basis 10 des Arguments.

LOG 100 ist gleich 2.

RND

1. RND numerischer ausdruck

Erzeugt Zufallszahlen. Ist das Argument kleiner 1, aber größer 0, so ist das Ergebnis kleiner 1 und größer oder gleich 0. Ist das Argument eine ganze Zahl und größer oder gleich 1, so ist das Ergebnis kleiner oder gleich dem Argument und größer oder gleich 1. Ist das Argument keine ganze Zahl und größer 1, so ist das Ergebnis kleiner oder gleich der nächstgrößeren ganzen Zahl des Arguments und größer oder gleich 1.

	Ergebnis	
Argument	Untergrenze	Obergrenze
.5	0<	<1
2	1	2
2.5	1	3

Die Sequenz ist jedes Mal die gleiche, wenn der PC 1260/1261 eingeschaltet wird, da die Basiszahl immer die gleiche ist. Um diese Basis-Zahl zufällig zu ändern, brauchen Sie den RANDOM-Befehl.

SGN

1. SGN numerischer ausdruck

Das Ergebnis ist abhängig vom Vorzeichen des Arguments. Ist das Argument positiv, ist das Ergebnis gleich 1; ist das Argument negativ, so ist das Ergebnis gleich -1; wenn das Argument gleich 0 ist, ist auch das Ergebnis gleich 0.

SGN -5 ist gleich -1.

SIN

1. SIN numerischer ausdruck

Errechnet den Sinus des Winkelarguments. Die Darstellung ist abhängig von dem Winkelmodus des PC 1260/1261 (DEG, RAD, GRAD).

SIN 30 ist im DEG-Modus gleich .5.

SQR

1. SQR numerischer ausdruck

Errechnet die Quadratwurzel des Arguments. Das Ergebnis ist identisch mit der Anwendung der speziellen [√]-Taste der Tastatur.

SQR 4 ist gleich 2.

TAN

1. TAN numerischer ausdruck

Errechnet den Tangens des Winkelarguments. Die Darstellung ist abhängig vom Winkelmodus des PC 1260/1261 (DEG, RAD, GRAD).

TAN 45 ist im DEG-Modus gleich 1.

STRING-FUNKTIONEN

Ein String ist eine Kette aus alphanumerischen und Sonderzeichen. String-Funktionen sind eine Gruppe von Operationen, die Strings manipulieren. Sie nehmen ein String und errechnen einen numerischen Wert dazu, oder Sie nehmen einen numerischen Wert und errechnen den String dazu, oder Sie bearbeiten den String als String. In vielen BASIC-Dialekten muß das Argument der Funktion in Klammern gesetzt werden. Nicht so bei dem PC 1260/1261. Klammern müssen nur dann gesetzt werden, wenn es nötig ist, das Argument von anderen Werten deutlich zu trennen. String-Ausdrücke mit zwei oder drei Ausdrücken benötigen immer eine Klammerung.

ASC

1. ASC "string ausdruck"

Errechnet den ASCII-Code des ersten Zeichens des Ausdrucks. Die ASCII-Zeichen-Code-Tabelle finden Sie in Anhang B.

```
ASC "A"   ist gleich 65.
```

CHR$

1. CHR$ numerischer ausdruck

Errechnet das ASCII-Zeichen des gegebenen numerischen Wertes. CHR$ ist die Umkehrfunktion der ASC-Funktion.

```
CHR$ 65   ist gleich "A".
```

LEFT$

1. LEFT$ ("string ausdruck",numerischer ausdruck)

Diese Funktion erstellt einen Teil-String des durch string ausdruck angegebenen strings. Von links beginnend wird die mit numerischer ausdruck gegebene Zahl von Zeichen herausgenommen.

```
LEFT$ ("ABCDEF",2)  ist gleich "AB".
```

LEN

1. LEN "string ausdruck"

Errechnet die Länge eines Strings.

```
LEN "ABCDE"    ist gleich 5.
```

MID$

1. MID$ ("string ausdruck",num asdr1, numasdr2)

Diese Funktion erstellt einen Teil-String aus dem durch ausdruck angegebenen string. Durch num asdr1 wird das erste Zeichen bestimmt, num ausdr2 bestimmt die Anzahl der herauszunehmenden Zeichen.

MID$ ("ABCDEF",2,3) ist gleich "BCD".

RIGHT$

1. RIGHT$ ("string ausdruck",numerischer ausdruck)

Diese Funktion erstellt einen Teil-String des durch string ausdruck angegebenen strings. Von rechts beginnend wird die Anzahl der von numerischer ausdruck gegebenenen Zeichen herausgenommen.

RIGHT$ ("ABCDEF",3) ist gleich "DEF".

STR$

1. STR$ numerischer ausdruck

STR$ wandelt einen numerischen Wert in einen String um.

STR$ 1.59 ist gleich "1.59".

VAL

1. VAL "string ausdruck"

VAL ist die Umkehrfunktion zu STR$. VAL wandelt einen String in einen numerischen Ausdruck um.

VAL "1.59" ist gleich 1.59.

Achtung: VAL kann nur numerische Zeichen (0 bis 9), Vorzeichen (+ und -) und das "E" für Exponenten umwandeln! Sind andere Zeichen in dem string ausdruck, so werden alle Zeichen rechts des letzten erlaubten Zeichens ignoriert. Spaces werden wie nicht-existent behandelt.

KOMMANDOFUNKTIONEN

Kommandofunktionen beziehen sich auf die Möglichkeit der Easy Simulation Programme (einfache Programm-Simulationen), kurz ESP. Sie sind mit dem BASIC des Rechners benutzbar. Die folgenden ESP-Kommandos werden beschrieben:

EQU = MEM =
LIST = NEW =
LLIST=

EQU =

1. EQU= numerischer ausdruck
2. EQU=

Abkürzung: EQ., EQU.
Vergleiche: NEW, NEW

Wirkung:
Verändert die Kapazität des ESP-Bereiches im Speicher.

Anwendung:
128 Bytes sind im Programmspeicher vorgesehen für ESP. Form 1 der Funktion ermöglicht es, die Kapazität des Bereiches zu verändern. Der Bereich kann in Einheiten (eine Einheit sind 128 Bytes) mit dem numerischen Ausdruck vorgegeben werden. Der ESP-Bereich ist also mit Ø gleich 128 Bytes, mit 1 gleich 256 Bytes, mit 2 gleich 384 Bytes usw. groß.

* Form 1 kann nur im PROgramm-Modus und mit neu initialisiertem Speicher, d. h. alle BASIC-Programme sind mit NEW gelöscht, benutzt werden.

* Wenn Sie die Funktion EQU= aufrufen, während ein ESP gespeichert ist und der Ausdruck der Funktion einen kleineren Bereich definiert, so erhalten Sie die Fehlermeldung ERROR 3.

* Wird der Bereich des ESP vergrößert, so verkleinern Sie damit den Programmdatenbereich.

ESP-Bereich (Standard)
ESP-Bereich (1. Erweiterung)
ESP-Bereich (2. Erweiterung)
ESP-Bereich (3. Erweiterung)
Programmdatenbereich Dieser Bereich wird mit jeder Erweiterung des ESP-Bereichs verkleinert.
Festwertebereich

Die zweite Form der Funktion zeigt Ihnen an, um wieviele Einheiten der ESP-Bereich erweitert wurde.

Achtung: Die zweite Form wird vom Rechner nicht mehr als Funktion, sondern als Kommando interpretiert. Ein numerischer Ausdruck darf nicht eingesetzt werden.

Beispiele:

EQU= 1	Der ESP-Bereich wird um 128 auf 256 Bytes erweitert.
EQU= 0	Der ESP-Bereich wird auf seinen Standard zurückgesetzt.
EQU=	Wird z. B. eine "3" ausgegeben, so ist der Bereich des ESP um 384 Bytes erweitert.

LIST=

1. LIST=
2. LIST= name

Abkürzung: L.=, LI.=, LIS.=
Vergleiche: LIST, LLIST=

Wirkung:
Gibt das gespeicherte ESP auf der Anzeige aus.

Anwendung:
Form 1 listet das erste gespeicherte ESP.

Form 2 listet das mit "name" angegebene ESP. Existiert kein ESP mit dem angegebenen Namen, so erhalten Sie die Fehlermeldung ERROR 4.

Beispiele:

LIST=	Das erste gespeicherte ESP wird gelistet.
LIST= total	Das erste ESP mit dem Namen "total" wird gelistet.

LLIST=

1. LLIST=
2. LLIST= name

Abkürzung: LL.=, LLI.=, LLIS.=
Vergleiche: LIST=, LLIST

Wirkung:
Die gespeicherten ESP werden ausgedruckt.

Anwendung:
Form 1 druckt alle gespeicherten ESP aus.

Form 2 druckt das ESP mit dem in "name" angegebenen Bezeichner aus. Existiert kein ESP mit diesem Namen, so erhalten Sie die Fehlermeldung ERROR 4.

* Diese Funktion kann nur in Verbindung mit dem Drucker/Mikrocassettenrecorder CE-125 benutzt werden.

Beispiele:

```
LLIST=          Druckt alle ESP aus.
LLIST= total    Druckt das ESP mit dem Namen "total" aus.
```

MEM

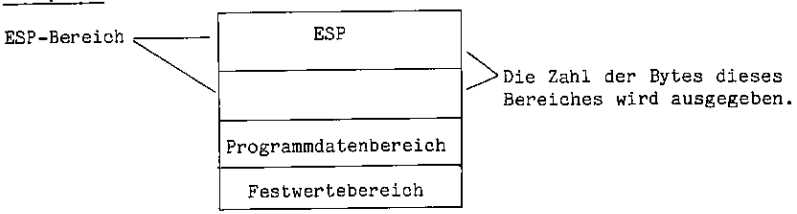
1. MEM=

Abkürzung: M.=, ME.=
Vergleiche: MEM, EQU=

Wirkung:
Gibt die Zahl der freien Bytes des ESP-Bereiches aus.

Anwendung:
Mit dieser Funktion können Sie überprüfen, wieviel Speicher Ihnen im ESP noch zur Verfügung steht. Sie erhalten als Ausgabe die Zahl der nicht belegten Bytes.

Beispiel:



NEW

1. NEW=

Vergleiche: NEW, EQU=

Wirkung:
Löscht das ESP.

Anwendung:
Alle ESP werden gelöscht, wenn Sie im PROgramm-Modus die Kommandofunktion NEW= eingeben. Der Umfang des mit der EQU=-Funktion gesetzten ESP-Bereiches bleibt unverändert.

KAPITEL 9

EASY SIMULATION PROGRAM (ESP)

Sollen nur Berechnungen mit dem PC 1260/1261 durchgeführt werden, kann das Programm in Form einer Gleichung ohne BASIC-Kommandos oder -Befehle eingegeben werden. Das Ergebnis ist ein Programm, in das Sie nur noch die für die Berechnung benötigten Werte eingeben müssen.

Dieses Kapitel beschreibt, wie solche ESP (Easy Simulation Programs) erstellt und benutzt werden.

ÜBERBLICK

Nehmen Sie an, Sie sollen A=B*C/2 berechnen. Sie müssen dafür die folgende Tastensequenz drücken:

Den Wert B * den Wert C / 2[ENTER]

Sind mehrere Ergebnisse mit verschiedenen Werten zu berechnen, so muß diese Sequenz jedes Mal wiederholt werden. Das BASIC-Programm dazu sieht so aus:

```
10: INPUT "B=";B;"C=";C
20: A=B*C/2
30: PRINT "A=";A
40: GOTO 10
```

Der Aufwand ist im Verhältnis zur Aufgabe recht hoch. Wesentlich einfacher wird es mit der Möglichkeit des ESP:

```
=A=B*C/2
```

Sie können die Gleichung direkt als Programm eingeben, indem Sie der Gleichung ein Nummernzeichen (=) voranstellen. Die Gleichung muß im PROgramm-Modus eingegeben werden. Wenn Sie jetzt in den RUN-Modus umschalten und

```
=A [ENTER]
```

eingeben, beginnt die Berechnung und Sie erhalten diese Ausgabe:

```
B   :C   :A
_
```

Wenn Sie die Werte für B und C eingegeben haben, wird das Ergebnis unter A ausgegeben. Ist der Wert für B z. B. gleich 17 und der Wert für C gleich 23, dann sieht der Ablauf folgendermaßen aus:

EINGABE	ANZEIGE
17 ENTER	B :C :A 17. :
23 ENTER	B :C :A 17. :23. :195.5

Nach einer weiteren Betätigung der ENTER -Taste beginnt das Programm von vorne. Sie brauchen also nur die verschiedenen Werte eingeben. Statt der einfachen Variablen "A", "B" und "C" können Sie auch einen längeren Variablennamen in der Gleichung benutzen, z. B.:

```
=INHALT=BASIS*HÖHE/2      (wobei BASIS die Grundseite eines
                           Dreiecks sei)
```

Die Variablennamen werden so angezeigt:

```
BASIS      :HÖHE      :INHALT
```

BESCHREIBUNG EINES ESP

Ein ESP hat folgendes Basisformat:

```
=variable = formel        (wobei formel = numerische Werte,
                           Variable und Operatoren sind)
```

Beispiele:

```
1)  =Y=2*X+3                        (y = 2x + 3)
2)  =INHALT = PI *RADIUS^2          (Kreisinhalt)
3)  =PREIS=kosten*(1+steuer/100)    (Preis=Kosten+Steuersatz)
```

Y, X, INHALT, kosten usw. sind in diesen Formeln die Variablen. Sie werden bei der Ausführung mit Werten belegt und das Ergebnis wird ausgerechnet und ausgegeben. Die Variablen der Formel und des ESP sind also gleich. Man unterscheidet aber zwischen Ein- und Ausgabevariablen.

* Ausgabevariable: Die Ausgabevariable steht auf der linken Seite des Gleichheitszeichens. Ihr wird das Ergebnis der Berechnung zugewiesen. Am Ende der Berechnung wird ihr Wert ausgegeben. In den Beispielen sind Y, INHALT und PREIS die Ausgabevariablen.

* Eingabevariable: Die Eingabevariablen stehen auf der rechten Seite des Gleichheitszeichen. Ihnen werden die für die Berechnung nötigen Werte zugewiesen. Der PC-1260/1261 fragt die Werte für die Variablen nacheinander ab. In den obigen Beispielen sind X, RADIUS, kosten und steuer die Eingabevariablen.

Wird in einem ESP die gleiche Variable links und rechts vom Gleichheitszeichen benutzt, so wird das Ergebnis der Berechnung gespeichert und im nächsten Durchgang in die Rechnung eingebracht (z. B.: ±HO=KO*HO - die Variable HO wird nicht als Eingabevariable abgefragt).

HINWEISE ZUR ERSTELLUNG EINES ESP

Rechen-Operatoren und Funktionen

1) Rechen-Operatoren
   + - * /
   Logische Operatoren wie <, > und <= können nicht benutzt werden.

2) Funktionen
   SIN COS TAN ASN ACS ATN LOG LN EXP
   SQR DMS DEG INT ABS SGN PI
   sowie die Funktionszeichen
   ^ √ ↑↑
   Logische Funktionen wie AND und NOT und die RND-Funktion können nicht benutzt werden.

3) Andere
   (,)

Format und Länge eines ESP

1) Am Anfang eines ESP muß immer ein Nummer-Zeichen (±) stehen. Es ist das Erkennungszeichen für den Rechner, daß dieses Programm ein ESP ist. Die Ausführung des ESP wird mit dem ± -Zeichen und dem Ausgabevariablennamen gestartet.
   Das ± -Zeichen muß und darf nur am Anfang eines ESP stehen.

2) In einem ESP darf nur die Ausgabevariable auf der linken Seite der Gleichung stehen. Steht ein Rechenkommando oder eine Funktion auf der linken Seite des Gleichheitszeichens, so erhalten Sie die Fehlermeldung ERROR 1.

3) Um mehrere Ausgabevariable mit verschiedenen Rechenvorgängen in einem ESP unterzubringen, müssen Sie diese mit Doppelpunkten trennen. Beispiel:

   ```
   ±A=A+1:B=2*A*A+4*A+1:C=2*B
   ```

4) Ein ESP beginnt mit dem ± -Zeichen und endet mit der [ENTER] - Taste. Die Länge eines ESP inklusive "±" und [ENTER] ist maximal 80 Bytes.

Variable

1) In einem ESP können bis zu 10 Variable benutzt werden. Davon steht eine, die Ausgabevariable, links und die restlichen, bis zu 9, rechts vom Komma als Eingabevariable. Werden mehr als 10 Variable benutzt, erhalten Sie die Fehlermeldung ERROR 3 bei der Ausführung.

2) Das erste Zeichen einer Variablen muß immer ein Buchstabe sein. Die folgenden Zeichen können sowohl Buchstaben als auch numerische Zeichen sein. Es dürfen Groß- und Kleinbuchstaben verwendet werden. Die Variablen dürfen keine BASIC-Kommandos, Funktionen oder Rechen-Operatoren enthalten, es sei denn, sie sind in Kleinbuchstaben geschrieben.

Beispiel:

```
=SINA=SIN A
=total=data1+data2
```

Das erste Beispiel gibt eine Fehlermeldung, da SINA einen Funktionsnamen, nämlich SIN, enthält. Das zweite Beispiel läuft, wenn im RUN-Modus das ESP mit Kleinbuchstaben aufgerufen wird.

3) Eine Variable darf maximal 7 Zeichen lang sein, alle weiteren Zeichen werden ignoriert.

4) Die Variablen eines ESP und eines BASIC-Programms sind unabhängig und werden getrennt gespeichert. Die Variable A eines ESP hat nichts zu tun mit der einfachen numerischen Variablen A eines BASIC-Programms. BASIC-Variable können also nicht in einem ESP benutzt werden.

ERSTELLUNG EINES EASY SIMULATION PROGRAM (ESP)

Wir wollen das Easy Simulation Program von Seite 160 in den Rechner eingeben. Gehen Sie nach den folgenden Anweisungen vor:

(1) Schalten Sie den PC 1260/1261 in den PROgramm-Modus.
(2) Geben Sie NEW= ein und drücken dann die [ENTER] -Taste. Damit löschen Sie den ESP-Speicherbereich. Dieser Schritt ist nicht nötig, wenn Sie die alte ESP bewahren wollen.

(3) Geben Sie jetzt das ESP ein.

EINGABE	ANZEIGE
=Y=2*X+3	=Y=2*X+3_
[ENTER]	=Y=2*X+3

Mit der [ENTER] -Taste übergeben Sie das ESP an den Rechner und der schreibt es in den ESP-Speicherbereich. Wenn Sie ein ESP mit einer Ausgabevariablen schreiben, die bereits für ein anderes ESP benutzt wurde, so wird das alte Programm gelöscht. Geben Sie also z. B. jetzt ein ESP

```
#Y=X*X
```

ein, wird das ESP

```
#Y=2*X+3
```

gelöscht. ESP mit gleicher Ausgabevariable können also nicht gleichzeitig gespeichert sein.

TESTEN DES EASY SIMULATION PROGRAMMS

Überprüfen Sie, ob Sie sich vertippt haben. Wenn ja, korrigieren Sie Ihren Tippfehler, bevor Sie die [ENTER] -Taste drücken. Wird ein ESP angezeigt, können Sie mit den Hoch- und Runter-Pfeilen weitere gespeicherte ESP zur Anzeige bringen.

Wenn kein ESP angezeigt wird, können Sie die ESP oder auch ein bestimmtes ESP mit dem LIST-Kommando ausgeben lassen.

DAS LIST# KOMMANDO

Mit diesem Kommando werden ESPs gelistet.

EINGABE:

LIST = [ENTER]

Die ESPs werden beginnend mit dem zuerst geschriebenen ausgegeben.

Die Form:

LIST#name

listet das ESP mit der Ausgabevariablen, die mit name bezeichnet wird. Ist kein entsprechendes ESP gespeichert, erhalten Sie die Fehlermeldung ERROR 4.

* Bei ESPs, die mehr als 48 Zeichen haben, müssen Sie die Rechts-Taste benutzten, um die weiteren Zeichen zur Anzeige zu bringen.

Wird bei dem Test eines ESP ein Fehler gefunden, korrigieren Sie diesen mit Hilfe der Editiermöglichkeiten des PC 1260/1261 (Kapitel 3).

Haben Sie eine Korrektur ausgeführt, die den Ausgabevariablennamen verändert hat, wird das Programm unter dem neuen Namen neu gespeichert. Das alte ESP bleibt erhalten.

Soll dieses gelöscht werden, müssen Sie erst

=name [ENTER] (wobei name=Ausgabevariable)

im PROgramm-Modus eingeben. Sie erhalten dann diese Anzeige:

```
=name          CLEAR
YES-1   OR NO-2:_
```

Geben Sie jetzt eine '1' ein, so wird das angegebene ESP gelöscht. Soll es doch nicht gelöscht werden, geben Sie eine '2' ein.

AUSFÜHRUNG DES EASY SIMULATION PROGRAMMS

ESPs werden im RUN-Modus ausgeführt. Sie müssen also als erstes Ihren PC 1260/1261 in den RUN-Modus umschalten. Dann geben Sie

=name [ENTER] (wobei name=Ausgabevariable)

ein. Wurde die Ausgabevariable des ESP bei der Programmierung in Kleinbuchstaben eingegeben, so muß das ESP im RUN-Modus auch mit Kleinbuchstaben aufgerufen werden. Wenn Sie das Beispiel

=Y=2*X+3

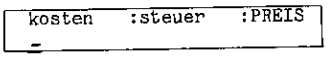
das Sie in den Rechner eingegeben haben, ausführen, sieht der Vorgang folgendermaßen aus:

EINGABE	ANZEIGE
=Y [ENTER]	X :Y _
7	X :Y 7_
[ENTER]	X :Y 7. :17.
[ENTER]	X :Y _

Es kann vorkommen, daß das Ergebnis auf der Anzeige nicht ausgegeben wird. Mit dem Rechts-Pfeil kann das Ergebnis dann in die Anzeige geholt werden. Benutzen Sie den Links-Pfeil, um wieder die alte Anzeige zu bekommen.

Wie Sie bei der Ausführung der Beispiele sehen konnten, werden die Variablennamen in der oberen Zeile wiedergegeben, die zugewiesenen Daten in der unteren.

Die Anzeige wird in drei Blöcken zu je 7 Zeichen aufgeteilt. Es können also drei Variablennamen dargestellt werden.



Werden also mehr als drei Variable in einer Gleichung benutzt, können nicht alle angezeigt werden. Die verdeckten Variablen erscheinen eine nach der anderen, wenn die jeweils ganz rechts stehende einen Wert zugewiesen bekommen hat.

Hat der Wert einer Variablen mehr als 7 Stellen, so wird ihr zusätzlicher Raum für die Darstellung eingeräumt. Es kann also vorkommen, daß nur zwei Variablen und ihre Werte auf der Anzeige wiedergegeben werden.

Sie sollten nach jeder Ausführung eines ESP mit dem Rechts-Pfeil überprüfen, ob Werte verdeckt sind. Ansonsten werden alle Werte normal wiedergegeben, es sei denn, daß die Anzeige mit einem USING-Befehl formatiert wurde.

Es kann auch nach einem USING-Befehl vorkommen, daß nicht alle Werte oder Stellen einer Variablen wiedergegeben werden können. Ein USING-Format muß vor der Ausführung eines ESP gegeben werden.

Beispiel:

EINGABE	ANZEIGE
USING "====="	USING"====="
[ENTER]	>
#PREIS [ENTER]	kosten :steuer :PREIS
5000 [ENTER]	kosten :steuer :PREIS 5000 :
5.55/100 [ENTER]	kosten :steuer :PREIS 5000 :0 :5277

Da die Anzeige mit dem USING-Befehl auf 6 Zeichen mit dem Dezimalpunkt ganz rechts formatiert wurde, kann 0.0555 nicht dargestellt werden. Ebenso können die Dezimalstellen des Ergebnisses nicht dargestellt werden.

* Während der Ausführung eines ESP kann der USING-Befehl gelöscht werden. Drücken Sie [SHIFT] und [CA] , um die Ausführung des Programms abzubrechen. Gleichzeitig wird auch das USING-Format gelöscht.

Wenn Sie während einer Eingabe die [CL] -Taste drücken, wird die letzte Eingabe gelöscht. Sie können die Ausführung eines ESP auch mit Umschalten in einen anderen Modus beenden.

KORREKTUR EINER EINGABE

In diesem Abschnitt wird die Korrektur einer Eingabe beschrieben. Ein Wert kann vor Drücken der [ENTER] -Taste oder nach Ausgabe des Ergebnisses korrigiert werden.

1) Korrektur vor dem [ENTER]

Drücken Sie die [CL] -Taste und geben Sie den korrekten Wert ein, oder benutzen Sie die Links- und Rechts-Pfeile, um an die falsche Stelle zu gelangen und korrigieren Sie dann diese Stelle.

Beispiel 1:

Der Wert von B soll 120 sein, es wurde aber 130 eingegeben. Das ESP sieht folgendermaßen aus: SUMME=A+B+C+D

Ausführung: RUN-Modus

EINGABE	ANZEIGE
SUMME [ENTER]	A :B :C
100 [ENTER] 130	A :B :C 100. :130_
◄ ◄	A :B :C 100. :1[3]0
2 [ENTER]	A :B :C 100. :120. :_

2) Sofortige Korrektur nach Beendigung der Eingabe

Die zweite Korrekturmöglichkeit haben Sie sofort nach Betätigen der [ENTER] -Taste:

EINGABE	ANZEIGE
135 [ENTER]	B :C :D 120. :135. :_

Der korrekte Wert sei 145.

◄	A :B :C 100. :120. :[1]35

Der Cursor wird an den Anfang der vorhergehenden Variablen, hier "C", gesetzt. Wurde bereits ein Wert für "D" eingegeben, bleibt der Cursor bei dem Wert der Variablen "D".

EINGABE	ANZEIGE
►	A :B :C 100. :120. :145
4 ENTER	B :C :D 120. :145. :

Der Wert von "C" ist auf 145 korrigiert.

3) Korrektur eines Wertes nach Ausgabe des Ergebnisses

Die dritte Möglichkeit der Korrektur eines Wertes haben Sie, nachdem das Ergebnis ausgegeben wurde. Bewegen Sie den Cursor mit der Links-Taste auf den zu korrigierenden Wert, ersetzen Sie ihn und drücken Sie die ENTER -Taste so oft, bis Sie das neue Ergebnis haben.

Beispiel:

Sie stellen fest, daß der Wert für "A" nicht 100, sondern 210 sein muß. Das Ergebnis wurde bereits ausgegeben. Wechseln Sie den Wert von "A" mit folgenden Schritten:

EINGABE	ANZEIGE
160 ENTER	C :D :SUMME 145. :160. :525.
◄◄◄◄	A :B :C 100. :120. :145.
210 ENTER	A :B :C 210. :120. :145.
ENTER	C :D :SUMME 145. :160. :635.

Da in diesem Beispiel der Wert von "B" nicht verändert wurde, wird sofort das neue Ergebnis ausgegeben. Soll auch der Wert von "C" korrigiert werden, müssen Sie den Cursor mit dem Rechts-Pfeil hinter den Wert von "B" bringen und dann die ENTER -Taste drücken. Der Cursor wird so an den Anfang der Variablen "C" gebracht.

Wird die ENTER -Taste ohne vorherige Dateneingabe gedrückt, so wird die Berechnung durchgeführt.

Achtung: Nach der Ausführung der Berechnung kann der Cursor mit dem Links-Pfeil auf jeden beliebigen Wert gesetzt werden. Mit dem Rechts-Pfeil kann die Position des Cursors nur innerhalb eines Wertes verändert werden.

LISTEN DER KOPFVARIABLEN

Die Ausgabevariable, also die Variable, die links des Gleichheitszeichens steht, wird auch Kopfvariable genannt. Sowohl beim Listing als auch bei Ausführung eines ESP wird es mit dieser Variablen angesprochen.

Es ist also, besonders wenn mehrere ESP gespeichert sind, sinnvoll, einen Befehl zur Verfügung zu haben, mit dem man sich diese Kopfvariablen listen kann.

Dieser Abschnitt beschreibt, wie diese Liste ausgegeben wird und wie man aus dieser Liste ein ESP startet.

(1) Listen der Kopfvariablen

Geben Sie

≠ ENTER

ein. Dieses Kommando kann sowohl im PROgramm-Modus als auch im RUN-Modus gegeben werden.

EINGABE	ANZEIGE
≠ ENTER	≠A ≠INHALT ≠Y ≠PREIS

Die Kopfvariablen werden in Gruppen zu sechst mit je 7 Zeichen und einem vorgestellten ≠-Zeichen ausgegeben. Sind im Speicher mehr als 6 ESP vorhanden, so bringen Sie die verdeckten mit dem Rechts-Pfeil zur Anzeige.

(2) Start eines ESP aus der Kopfvariablenliste

Wenn Sie im RUN-Modus das ≠-Kommando gegeben haben, können Sie mit den Links- und Rechts-Pfeilen den Cursor auf eine der Kopfvariablen setzen. Drücken Sie jetzt ENTER und das ESP wird gestartet.

Beispiel: RUN-Modus

EINGABE	ANZEIGE
≠ ENTER	≠A ≠INHALT ≠Y ≠PREIS
►	≠A ≠INHALT ≠Y PREIS
► ►	≠A ≠INHALT ≠Y ≠PREIS
ENTER	X :Y _

(3) Listen eines ESP aus der Kopfvariablenliste

Wenn Sie im PROgramm-Modus das ≠-Kommando gegeben haben, so können Sie mit den Rechts- und Links-Pfeilen den Cursor auf eine bestimmte Kopfvariable bringen. Drücken Sie jetzt die ENTER -Taste, um dieses ESP zu listen.

Beispiel: PRO-Modus

EINGABE	ANZEIGE
≠ ENTER	≠A ≠INHALT ≠Y ≠PREIS
► ► ►	≠A ≠INHALT ≠Y ≠PREIS
ENTER	≠Y=2*X+3 ≠PREIS=kosten*(1+steuer/

Löschen eines ESP

Es gibt zwei Möglichkeiten, ESPs zu löschen.

(1) Alle ESPs löschen: Um alle ESPs im Speicher zu löschen, geben Sie den Befehl:

NEW≠ ENTER

(2) Löschen eines einzelnen ESP: Mit folgenden Schritten können Sie ein einzelnes ESP löschen:

1. Den PC 1260/1261 in den PRO-Modus schalten.
2. Die Kopfvariable eingeben: ≠ name ENTER
   wobei name = Kopfvariable

3. Beantworten Sie die Meldung des PC 1260/1261 mit

1 [ENTER] um das ESP zu löschen
2 [ENTER] um das ESP NICHT zu löschen.

Beispiel: PRO-Modus

=Y soll gelöscht werden.

EINGABE	ANZEIGE
=Y [ENTER]	=Y CLEAR YES-1 OR NO-2:_
1 [ENTER]	>

Das ESP =Y ist jetzt gelöscht.

* Existiert kein ESP mit der angegebenen Kopfvariablen, so kehrt der Rechner direkt in den PRO-Modus zurück.

AUSDRUCK WÄHREND DER AUSFÜHRUNG EINES ESP

Wenn an dem PC 1260/1261 ein Drucker/Mikrocassettenrecoder CE-125 angeschlossen ist, können die Variablen und ihre Werte ausgedruckt werden.

Drücken Sie erst die [SHIFT] -Taste, dann die [P ↔ NP]-Taste. Ein kleiner Balken wird über dem Schriftzug "PRINT" unter der Anzeige erscheinen. Der PC 1260/1261 ist in den PRINT-Modus umgeschaltet. Wiederholung der Tastenfolge schaltet den PRINT-Modus wieder aus.

Wird jetzt ein ESP ausgeführt, so werden alle Variablen und die zugewiesenen Werte über den Drucker ausgegeben.

* Sind Variable oder Werte, z. B. durch einen USING-Befehl, zu lang für eine Zeile, so gibt der Drucker zwei Zeilen aus.

DIE KOPFVARIABLEN AUSDRUCKEN

Um die Kopfvariablen auszudrucken, muß der CE-125 am PC 1260/1261 angeschlossen sein. Schalten Sie den Rechner jetzt wie oben beschrieben in den PRINT-Modus und geben dann

= [ENTER]

ein, werden die Kopfvariablen auf dem Drucker ausgegeben.

EIN EASY SIMULATION PROGRAMM AUSDRUCKEN

Ein Easy Simulation Programm kann mit dem LLIST=-Kommando ausgedruckt werden. Schließen Sie zuerst den CE-125 an den PC 1260/1261 an.

Geben Sie jetzt das Kommando:

LLIST= [ENTER]

oder

LLIST= name [ENTER] (wobei name = Kopfvariable)

Die erste Form druckt alle ESP in der Reihenfolge ihrer Erstellung aus, die zweite Form nur das mit name angegebene ESP.

KAPITEL 10

HELP-FUNKTIONEN

Es wird Ihnen sicherlich gelegentlich geschehen, daß Sie sich an einen Programmbefehl oder seine Verwendung nicht erinnern können, wenn Sie gerade ein Programm schreiben.

In solchen Momenten werden Sie im Handbuch nach dem betreffenden Befehl suchen, sofern es in Reichweite ist - aber dies ist ein mühsames Unterfangen.

Der Computer ist in der Lage, Funktionen und ihre typischen Schreibweisen aufzulisten (Referenz-Funktion).

Darüber hinaus ist eine Zeichen-Code-Tabelle eingebaut, die ebenfalls aufgelistet werden kann (ASCII-Tabelle). Auch wenn ein Fehler aufgetreten ist, kann der Fehlertyp angezeigt werden.

Dieser Abschnitt beschreibt, wie Sie vorgehen, wenn Sie bestimmte Informationen auflisten lassen wollen.

REFERENZ-FUNKTION

Es gibt zwei Möglichkeiten, die Referenz-Funktion zu benutzen.

(1) Wenn Sie einen Befehl oder seine Schreibweise vergesssen haben, können Sie die Befehle des Computers auflisten lassen und diese betrachten.
Es können entweder alle Befehle abgerufen werden oder, wenn Sie nur den ersten (oder die ersten beiden) Buchstaben eingeben, die mit dem jeweiligen Buchstaben beginnenden Befehle.

(2) Wenn Sie die Schreibweise eines Befehls vergessen haben, kann eine Muster-Schreibweise des Befehls abgerufen werden.
Zu diesem Zweck können Sie den Befehl zunächst mit Methode (1) suchen oder den Befehlsnamen direkt eingeben.

Das Auflisten von BASIC-Befehlen und -Funktionen

Um die Befehle des Computers von Anfang an in alphabetischer Reihenfolge listen zu lassen, geben Sie ein:

HELP
SHIFT 7

Nun werden die mit A beginnenden Befehle ausgegeben. Um die weiteren Befehle zu erhalten, betätigen Sie die ↓-Taste. (Wollen Sie diese Funktion in der Mitte einer Eingabesequenz abrufen, drücken Sie die CL -Taste, dann SHIFT und HELP nacheinander.

EINGABE ANZEIGE

[CL] [SHIFT] [HELP]

```
ABS     ACS     AND
AREAD   ASC     ASC
```

[↓]

```
AREAD   ASC     ASN
ATN
```

Weiterhin kann ein Befehl ausgegeben werden, indem Sie seinen ersten oder seine ersten beiden Buchstaben spezifizieren (eingeben). Drücken Sie beispielsweise C und dann nacheinander [SHIFT] und [HELP], werden die mit "C" beginnenden Befehle angezeigt.

Anmerkung: Während Sie Daten über die Referenz-Funktion anzeigen lassen, lassen sich nur die Tasten [↓], [↑] und [CL] bedienen. Darum löschen Sie, wenn Sie ein Zeichen eingeben wollen, die Referenz-Anzeige zunächst mit der [CL] -Taste.

EINGABE ANZEIGE

[CL] L [SHIFT] [HELP]

```
LEFT$   LEN     LET
LIST    LIST=   LLIST
```

[↓]

```
LIST    LIST=   LLIST
LLIST=  LN      LOG
```

.
.
.
.
.

[CL] RE [SHIFT] [HELP]

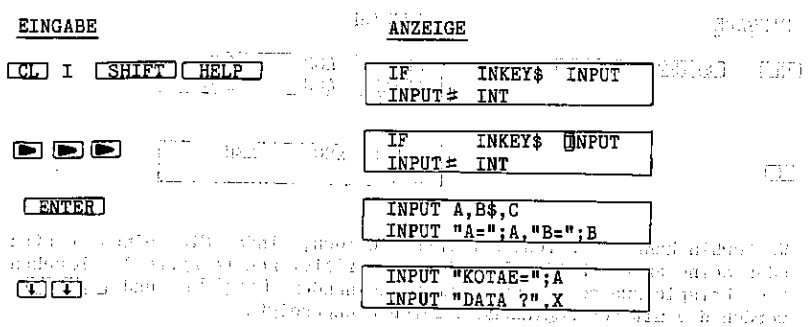
```
READ    REM     RESTORE
RETURN
```

Anmerkung: Wenn es keinen Befehl gibt, der mit dem eingegebenen Buchstaben beginnt, wird kein Befehl angezeigt werden, auch wenn Sie [SHIFT] und [HELP] nacheinander drücken.

Die Ausgabe der Musterschreibweise eines Befehls

Nachdem Sie den gesuchten Befehl mit Methode (1) gefunden haben, bewegen Sie den Cursor mit Hilfe des [▶] -Pfeiles auf den Anfang des Befehls und betätigen dann die [ENTER] -Taste. So erhalten Sie die Musterschreibweise für den betreffenden Befehl.

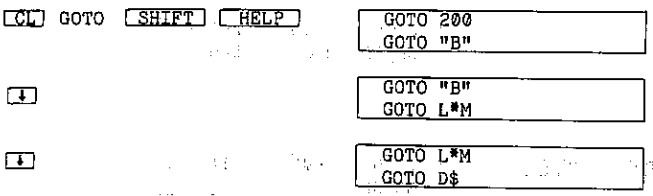
Lassen Sie die Musterschreibweise für den INPUT-Befehl auflisten.



Anmerkung: Wenn es drei Musterschreibweisen gibt (wie im obigen Beispiel), können nach dem Auflistungsbefehl durch ENTER die anderen Muster mit Hilfe der ↓-Taste abgerufen werden. Daneben kann nach Eingabe des Befehls die Musterschreibweise auch durch Betätigen der Tasten SHIFT und HELP abgefragt werden.

Beispiel:

Lassen Sie die Musterschreibweise für den GOTO-Befehl auflisten.



ASCII-TABELLE

Der Computer ist mit einer eingebauten Zeichen-Code-Tabelle ausgestattet. Diese Tabelle können Sie auflisten lassen, indem Sie eingeben:

ASCII SHIFT HELP

Mit den ↓- und ↑-Tasten können Sie sich in der Tabelle auf- und abwärts bewegen.

Die in diesem Fall ausgegebenen Zeichen (Buchstaben, Zahlen, Symbole etc.) können mit dem CHR$-Befehl spezifiziert werden.

Beispiel:

EINGABE ANZEIGE

[CL] ASCII [SHIFT] [HELP]

```
   0 1 2 3 4 5 6 7 8 9 A B C D E F
2 :   ! " # $ % & ' ( ) * + , - . /
```

[↓]

```
   0 1 2 3 4 5 6 7 8 9 A B C D E F
3 : 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
```

. . .

[↓]

```
   0 1 2 3 4 5 6 7 8 9 A B C D E F
7 : p q r s t u v w x y z
```

Wie im Beispiel gezeigt, wird die niedrigere Stelle des Hexadezimalcodes in der oberen Zeile der Anzeigeeinheit ausgegeben, die höhere Stelle des Hexadezimalcodes erscheint in der unteren Zeile. Die Zeichen werden im direkten Anschluß an den Doppelpunkt (:) ausgegeben. Der Zeichen-Code wird wie folgt bestimmt:

Beispiel:

Der Code für * ist 2A.

```
   0 1 2 3 4 5 6 7 8 9 A B C D E F
2 :   ! " # $ % & ' ( ) * + , - . /
```

Anmerkung: Der Code wird hier in hexadezimaler Schreibweise wiedergegeben. (2A ist in dezimaler Schreibweise 42.)

Wenn Sie ein Zeichen mit CHR$ spezifizieren wollen, geben Sie es in dezimaler Schreibweise an oder benutzen den &-Befehl. Wandeln Sie die hexadezimale Schreibweise in dezimale um und spezifizieren Sie Ihr Zeichen erst dann.

Beispiel:

Wenn Sie * spezifizieren wollen:

```
CHR$42    (dezimale Schreibweise)
CHR$&2A   (hexadezimale Schreibweise, durch & in dezimale
           umgewandelt)
```

Die Zeichen-Code-Tabelle finden Sie in ANHANG B

FEHLERMELDUNGSFUNKTION

Wenn während des Rechnens ohne Programmunterstützung ein Fehler auftritt, können Sie durch Betätigen der [SHIFT] - und der [HELP] - Taste den Fehler lokalisieren lassen und sich eine Erklärung des Fehlertyps geben lassen.

Tritt ein Fehler während eines Programmablaufs auf, drücken Sie auf [SHIFT] und [HELP], dann wird die Programmzeile, in der der Fehler zu finden ist, ausgegeben. Zusätzlich erhalten Sie Information über den Fehlertyp.

Beispiel:

Bringen Sie den Schiebeschalter in die RUN-Position.

EINGABE	ANZEIGE
[CL] 2 * / 3 [ENTER]	ERROR 1
[SHIFT] [HELP]	2 * / 3 SYNTAX ERROR

Der blinkende Cursor oben in der Anzeige gibt die Stelle an, an der der Fehler liegt. Die untere Zeile informiert Sie über den Fehlertyp.

Beispiel:

Bringen Sie den Schiebeschalter in die PROgramm-Stellung.

EINGABE	ANZEIGE
[NEW] [ENTER]	>
10 A = 5 / 0 + 3 [ENTER]	10: A = 5 / 0 + 3
20 PRINT A [ENTER]	20: PRINT A

Bringen Sie den Schiebeschalter in die RUN-Stellung.

RUN [ENTER]	ERROR 2 in 10

Die Anzeige läßt Sie wissen, daß der Fehler in Zeile 10 liegt. Wollen Sie herausfinden, wo genau in der Zeile 10 und möchten Sie auch den Fehlertyp wissen, betätigen Sie [SHIFT] und [HELP] nacheinander.

EINGABE	ANZEIGE
SHIFT HELP	10: A = 5 / 0 + 3 CALCULATION ERROR

Im Display wird nun angezeigt, daß die 0 in Zeile 10 ein Rechenfehler ist. Bringen Sie den Schiebeschalter wieder in die PROgramm-Stellung und betätigen Sie nun die ↑- oder ↓ -Taste.

Geben Sie nun beispielsweise 8 ein und drücken Sie ENTER . Bringen Sie den Schiebeschalter in die Position RUN, geben Sie RUN ein und drücken Sie wieder ENTER .

EINGABE	ANZEIGE
RUN	3.625

Dies ist das Ergebnis der Rechenoperation 5/8+3.

KAPITEL 11

PANNENHILFE

In diesem Kapitel sollen Sie einige Hinweise erhalten, was Sie unternehmen können, wenn Ihr SHARP PC 1260/1261 nicht tut, was Sie von ihm erwarten. Es ist in zwei Teile gegliedert - der erste beschäftigt sich mit allgemeiner Bedienung des Gerätes, und der zweite mit der BASIC-Programmierung. Für jedes Problem schlagen wir eine Reihe von Lösungen vor. Sie sollten jede von ihnen versuchen - aber nur eine zur Zeit -, bis Sie das Problem gelöst haben.

BEDIENUNG DES GERÄTS

Wenn:	Dann sollten Sie:
Sie das Gerät einschalten, aber nichts in der Anzeige erscheint	1) Überprüfen, ob der Schiebeschalter sich in der RUN-, PRO- oder RSV-Position befindet. 2) Die ON-/BRK-Taste betätigen, um festzustellen, ob sich das Gerät automatisch abgeschaltet hatte. 3) Die Batterien wechseln. 4) Den Kontrast einstellen.
Die Anzeige zwar funktioniert, aber keine Reaktion auf Tastendruck erfolgt.	1) [CL] drücken, um die Anzeige zu löschen. 2) [SHIFT] [CA] drücken, um die Anzeige zu löschen. 3) Das Gerät aus- und wieder einschalten. 4) Irgendeine Taste drücken, festhalten und den ALL-RESET-Schalter drücken. 5) Den ALL-RESET-Schalter ohne zusätzlichen Tastendruck betätigen.
Sie eine Rechnung oder Antwort eingeben, aber keine Reaktion erfolgt.	1) Drücken Sie [ENTER] .
Sie ein BASIC-Programm abarbeiten lassen, etwas angezeigt wird und das Programm dann anhält.	1) Drücken Sie [ENTER] .

Sie eine Rechnung eingeben und diese in der Form eines BASIC-Statements angezeigt wird (Doppelpunkt nach der ersten Zahl).	1) Schalten Sie vom PROgramm- in den RUN-Modus um.
Sie keinerlei Reaktion auf Tastenbetätigung erhalten.	1) Halten Sie irgendeine Taste fest und drücken Sie den ALL-RESET-Schalter. 2) Wenn Sie dann noch immer keine Reaktion erhalten, drücken Sie den ALL-RESET-Schalter, ohne dabei eine andere Taste festzuhalten. Dabei werden jedoch alle Daten, Programme und Speicherinhalte gelöscht.

FEHLERSUCHE IM BASIC

Wenn Sie ein neues BASIC-Programm eingeben, wird dieses in der Regel beim ersten Startversuch nicht laufen. Selbst, wenn Sie nur ein Programm abtippen, von dem Sie wissen, daß es korrekt ist, wie z. B. die in diesem Handbuch vorgestellten, dürfte Ihnen normalerweise mindestens ein Tippfehler unterlaufen. Handelt es sich um ein längeres Programm, wird es oft auch mindestens einen logischen Fehler enthalten. Es folgen einige grundsätzliche Hinweise, wie Sie solche Fehler finden und korrigieren.

Sie starten Ihr Programm und erhalten eine Fehlermeldung:

1. Schalten Sie zurück in den PROgramm-Modus und benutzen Sie die [↑]- oder [↓]-Taste, um die fehlerhafte Zeile ins Display zu rufen. Der Cursor wird sich an der Stelle befinden, wo Ihr PC 1260/1261 verwirrt wurde.

2. Wenn Sie aus der Art, wie die Programmzeile geschrieben ist, keinen offensichtlichen Fehler entnehmen können, kann das Problem auch an den von Ihnen verwendeten Werten liegen. So wird beispielsweise CHR$(A) eine Fehlermeldung bewirken, wenn A den Wert 1 hat, weil CHR$(1) ein unerlaubtes Zeichen ist. Überprüfen Sie die Werte der von Ihnen verwendeten Variablen, indem Sie entweder im RUN- oder im PROgramm-Modus die einzelnen Variablennamen gefolgt von [ENTER] eingeben.

Sie starten das Programm mit RUN und erhalten keine Fehlermeldung, doch das Programm tut nicht, was Sie von ihm erwarten:

1. Überprüfen Sie das Programm Zeile für Zeile unter Verwendung von LIST und den [↑]- und [↓]-Tasten, um herauszufinden, ob Sie das Programm korrekt eingegeben haben. Es ist erstaunlich, wieviele Fehler beim bloßen erneuten Durchsehen eines Programms gefunden werden können!

2. Versuchen Sie, jede einzelne Zeile beim Durchsehen so zu interpretieren, als wären Sie ein Computer. Nehmen Sie einfache Werte und realisieren Sie die Operationen der einzelnen Zeilen, um herauszufinden, ob Sie die gewünschten Ergebnisse erhalten.

3. Fügen Sie eines oder mehrere zusätzliche PRINT-Statements in Ihr Programm ein, um die einzelnen Werte und Tastenbelegungen zur Anzeige zu bringen. Benutzen Sie diese, um die korrekten Teile des Programms von den möglicherweise fehlerhaften zu isolieren. Diese Vorgehensweise ist auch nützlich um zu bestimmen, welche Teile des Programms schon abgearbeitet wurden. Sie können den Programmablauf auch an kritischen Stellen vorübergehend mit STOP unterbrechen, um dann einzelne Variable zu überprüfen.

4. Verwenden Sie TRON und TROFF, entweder als Befehle oder als Programmbestandteile, um den Programmablauf durch die einzelnen Zeilen hindurch verfolgen zu können. Halten Sie das Programm an kritischen Punkten an, um den Inhalt von wichtigen Variablen zu überprüfen. Dies ist zwar eine sehr langsame Methode, Fehler aufzuspüren - aber es ist manchmal die einzige.

KAPITEL 12

DIE INSTANDHALTUNG DES PC 1260/1261

Um reibungsloses Funktionieren Ihres PC 1260/1261 zu gewährleisten, empfehlen wir die Beachtung der folgenden Punkte:

- Gehen Sie immer vorsichtig mit Ihrem Taschencomputer um, da die Flüssigkristallanzeige aus Glas gefertigt ist.
- Halten Sie den Computer von starken Temperaturschwankungen fern, ebenso von Feuchtigkeit oder Staub. Wenn Sie bei warmem Wetter Ihren PKW längere Zeit der direkten Sonne aussetzen und sich hohe Temperaturen aufbauen, kann das Ihren Computer beschädigen!
- Benutzen Sie zur Reinigung Ihres Computers ausschließlich ein trockenes, weiches Tuch. Verwenden Sie niemals Lösungsmittel, Wasser oder ein feuchtes Tuch!
- Um das Auslaufen der Batterien zu vermeiden, entfernen Sie diese, wenn Sie den Computer für längere Zeit nicht benutzen wollen.
- Wenn Sie die Hilfe einer Werkstatt benötigen, geben Sie Ihren Computer nur in ein autorisiertes SHARP Service Center!
- Wenn der Computer starker statischer Aufladung oder auch starkem Lärm ausgesetzt ist, kann er sich unter Umständen "aufhängen" (d. h. auf Tastendruck nicht mehr reagieren). Falls dies auftritt, drücken Sie den ALL-RESET-Schalter und halten dabei eine Taste fest (siehe Pannenhilfe).
- Bewahren Sie dieses Handbuch gut auf, falls Sie später einmal etwas nachschlagen wollen.

(Anmerkung: Näheres zur Instandhaltung der Option CE-125 finden Sie in Kapitel 7).

PROGRAMMBEISPIELE

Sie haben sich jetzt mit den Programmierungs-Kommandos und -Befehlen des PC 1260/1261 bekannt gemacht. Es ist nun wichtig, daß Sie Programme unter Verwendung des Handbuches erstellen. Wie beim Autofahren oder beim Tennisspiel können Sie sich mit der Übung unter Anleitung die nötige Praxis erwerben, eigene Programme zu schreiben. Die folgenden Seiten bieten Ihnen deshalb eine Auswahl von Programmen in ESP und BASIC an.

(SHARP Corporation und/oder ihre Vertretungen gewähren keine Haftung für Verluste oder Schäden, die aus der Verwendung dieser Programme entstehen könnten.)

INHALT

1. EASY SIMULATION PROGRAMME (ESP)
* Einleitung
* Golf-Scoring
* Zwischensumme und Endsumme
* Berechnung der Monatsraten
* Preis-Rabatt-Liste
* Kosten-Neutralpunkt-Analyse
* Schulnoten-Schnitt

2. BASIC-Anwender-Programme
* Nullstellen einer Funktion
* Mittelwert, Varianz und Standardabweichung
* Schnitt zwischen Kreisen und Geraden
* Berechnung der Anzahl von Tagen zwischen zwei Daten
* Schreibübungen
* Mondlandung
* Gedächtnistest
* Käferjagd
* Doppelte Rotation

EASY SIMULATION PROGRAMME (ESP)

EINLEITUNG

Die Beschreibung und Erklärung der ESPs setzen voraus, daß die Formeln für alle sechs Programme bereits in den PC 1260/1261 eingegeben wurden. Im folgenden wird die Eingabe der Formeln als ESP und deren Anwendung beschrieben.

Der benötigte Speicher für jede der Formeln ist kleiner als 80 Bytes, d. h. 480 Bytes müssen für die sechs Formeln bereitgestellt werden. Wir müssen also den ESP-Speicherbereich um drei Blöcke vergrößern, denn

(480-128)/128=2.75 < 3.

Eingabe der Formeln

(1) Schalten Sie den Rechner in den PROgramm-Modus.
(2) Geben Sie NEW [ENTER] und NEW= [ENTER] ein.
(3) Geben Sie EQU=3 [ENTER] ein.
(4) Geben Sie die Formeln ein:

```
=GrsScor=Outscor+Inscor:
 Netscor=GrsScor-Hand.          Golf-Scoring

=ZwSum=StkPr+Menge: Gesam
 t=Gesamt+ZwSum                 Zwischensumme und Endsumme

=MZs=1+(JZs/12)/100:RatZ
 hl=( LOG Rate- LOG (
 ABS (Kredit*(MZs-1)-Rat
 e)))/ LOG MZs                  Berechnung der Monatsraten

=Rbt10%Preis*.9:Rbt15%=
 Preis*.85:Rbt20%=Preis*
 .8                             Preis-Rabatt-Liste

=VKstRt=VarKstn/Verk:Kst
 nNPk=FixKstn/(1-VKstRt)        Kosten-Neutralpunkt-Analyse

=ZensSum=kunst+engl+math
 e+physik+sport:SCHNITT=
 ZenSum/5                       Schulnoten-Schnitt
```

Ausführung eines ESP

Schalten Sie den PC 1260/1261 in den RUN-Modus und geben Sie dann ein:

= [ENTER]

Sie erhalten eine Liste der ESP. Setzen Sie jetzt den Cursor auf das gewünschte Programm und drücken Sie die [ENTER] -Taste.

Beispiel:

Eingabe	Anzeige
=ENTER	=GrsScor=ZwSum =MZs =Rbt10% =VKstRt =ZenSum
►►	=GrsScor ZwSum =MZs =Rbt10% =VkstRt ZenSum
ENTER	StkPr :Menge :ZwSum _

GOLF-SCORING

Befreien Sie sich von schwerfälliger Punktezählung. Die Formel dieses Beispiel-ESP errechnet Ihnen die "Net-Scores" jedes einzelnen Golfspielers, basierend auf "Out-Score", "In-Score" und "Handicap".

Beispiel:

Name	Out-Score	In-Score	Handi-cap	Gros-Score	Net-Score
J. Müller	50	48	5	98	93
E. Schmidt	55	52	10	107	97
H. Meier	43	50	2	93	91

Berechnung:

"Gros-Score"="Out-Score"+"In-Score"
"Net-Score"="Gros-Score"-"Handicap"

Eingabe der Formel:

```
=GrsScor=Outscor+Inscor:
 Netscor=GrsScor-Handi
```

Anwendungssequenz:

	Tastenfolge	Anzeige	Bemerkung
1	[SHIFT] [CA]	>	Initialisierung
2	= [ENTER]	=GrsScor=ZwSum =MZs =Rbt10% =VkstRt =ZensSum	Formel Menü
3	[ENTER]	OutScor:Inscor :Handi _	Out-Score Abfrage
4	50 [ENTER]	Outscor:Inscor :Handi 50. :_	In-Score Abfrage
5	48 [ENTER]	Outscor:Inscor :Handi 50. :48. :_	Handicap Abfrage
6	5 [ENTER]	Inscor :Handi :GrsScor 48. :5. :98.	Ausgabe Gros-Score
7	[▶]	Handi :GrsScor:Netscor 5. :98. :93	Anzeige verschieben
8	[ENTER]	Outscor:Inscor :Handi _	Wiederholung

ZWISCHENSUMME UND ENDSUMME

Sie können auch kumulierende Formeln als ESP in den PC 1260/1261 eingeben. Dieses ESP errechnet Ihnen die Zwischensummen verschiedener Artikel aus Preis und Menge. Diese Zwischenergebnisse werden zu einer Endsumme aufaddiert.

Beispiel:

Modell	Stückpreis	Menge	Zwischensumme	Gesmtsumme
XC-1240	150	20	3000	3000
XD-1250	300	30	9000	12000
XE-1500	600	40	24000	36000
Endsumme			36000	

Die Formeln

```
Zwischensumme=Stückpreis*Menge
Endsumme = Summe der Zwischensummen
```

Eingabe der Formeln als ESP

```
ZwSum = Zwischen-Summe
StkPr = Stück-Preis

=ZwSum=StkPr*Menge:Gesam
 t=Gesamt+ZwSum
```

Anwendungssequenz

	Tastenfolge	Anzeige	Bemerkung
1	SHIFT CA	>	Initialisierung
2	=ENTER	=GrsScor=ZwSum =MZs =Rbt10% =VkstRt =ZensSum	Formel Menü
3	►► ENTER	StkPr :Menge :ZwSum _	Stückpreis Abfrage
4	150 ENTER	StkPr :Menge :ZwSum 150. :_	Menge Abfrage
5	20 ENTER	StkPr :Menge :ZwSum 150. :20. :3000.	Ausgabe Abfrage
6	►	Menge :ZwSum :Gesamt 20. :3000. :3000.	Ausgabe Abfrage
7	ENTER	StkPr :Menge :ZwSum _	Abfrage 2. Stückpreis
8	300 ENTER	StkPr :Menge :ZwSum 300. :_	Abfrage 2. Menge
9	30 ENTER	StkPr :Menge :ZwSum 300. :30. :9000.	Ausgabe 2. Zwischensumme
10	►	Menge :ZwSum :Gesamt 30. :9000. :12000.	Ausgabe Endsumme
11	ENTER	StkPr :Menge :ZwSum _	Wiederholung

BERECHNUNG DER MONATSRATEN

Erstellen Sie sich einen günstigen Abzahlungsplan. Für ein Darlehen sollte man sich einen vernünftigen Abzahlungsplan machen. Dieses Beispiel-ESP errechnet die Anzahl der nötigen monatlichen Raten abhängig von den Rückzahlungsmöglichkeiten.

Beispiel:

Das folgende Beispiel geht von einem Kredit in Höhe von DM 10.000,-- aus, zu 14 % Jahreszins und mit einer Monatsrate von DM 200,-.

Kredit = 10000 Monatsrate = 200
Jahreszins = 14 % (Monatszins = Jahreszins/12)

Berechnung:

Monatszins = 1 + (Jahreszins/12)/100

$$\text{Ratenzahl} = \frac{\log \text{Rate} - \log(\text{Kredit} * (\text{Monatszins} - 1) - \text{Rate})}{\log \text{Monatszins}}$$

Eingabe der Formel:

Monatszins = MZs Jahreszins = JZs Ratenzahl = RatZhl

```
=MZs=1+(JZs/12)/100:RatZ
 hl=( LOG Rate- LOG (
 ABS (Kredit*(MZs-1)-Rat
 e)))/ LOG Mzs
```

Anwendungssequenz

	Tastenfolge	Anzeige	Bemerkung
1	SHIFT CA	>	Initialisierung
2	= ENTER	=GrsScor=ZwSum =MZs =Rbt10% =VkstRt =ZensSum	Formel Menü
3	▶ ▶ ▶	=GrsScor=ZwSum =MZs =Rbt10% =VkstRt =ZensSum	Formel wählen
4	ENTER	JZs :Rate :Kredit _	Abfrage Jahreszins
5	14 ENTER	JZs :Rate :Kredit 14. :_	Abfrage Rate
6	200 ENTER	JZs :Rate :Kredit 14. :200. :_	Abfrage Kredit
7	10000 ENTER	Kredit :MZs :RatZ 10000. :1.011666667:75.4	Ausgabe Monatszins
8	▶	MZs :RatZhl 1.011666667:75.47706694	Ausgabe Ratenzahl

PREIS-RABATT-LISTE

Warum machen Sie sich nicht eine Preisliste zu den verschiedenen Angeboten? Es ist nicht immer möglich, einen Sonderpreis abzuwarten. Also sollte man die Preise vergleichen und sich eine Liste der Angebote machen.

Beispiel:

Das folgende Beispiel erstellt eine Sonderpreis-Liste mit 10 %, 15 % uns 20 % Rabatt für zwei Maschinen, die DM 1.698,-- und DM 1.980,-- kosten sollen.

Eingabe der Formel:

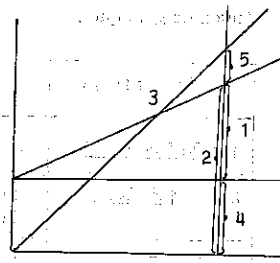
```
10 % = Rbt10%
15 % = Rbt15%
20 % = Rbt20%

    =Rbt10%=Preis*.9:Rbt15%=
     Preis*.85:Rbt20%=Preis*
     .8
```

Anwendungssequenz

	Tastenfolge	Anzeige	Bemerkung
1	SHIFT CL	>	Initialisierung
2	= ENTER	=GrsScor=ZwSum =MZs =Rbt10% =VkstRt =ZensSum	Formel Menü
3	▶ ▶ ▶ ▶	=GrsScor=ZwSum =MZs =Rbt10% =VkstRt =ZensSum	Formel wählen
4	ENTER	Preis :Rbt10% :Rbt15% _	Abfrage Preis
5	1698 ENTER	Preis :Rbt10% :Rbt15% 1698. :1528.2 :1443.3	Ausgabe 10 % und 15 % Nachlaß
6	▶ ▶	Rbt10% :Rbt15% :Rbt20% 1528.2 :1443.3 :1358.4	Ausgabe 10 %, 15 % und 20 %
7	ENTER	Preis :Rbt10% :Rbt15% _	Wiederholung

KOSTEN-NEUTRALPUNKT-ANALYSE

Wir wollen den Kosten-Neutralpunkt analysieren. Die Abbildung stellt den Kosten-neutralpunkt und die nötigen Daten dar. Das folgende Beispiel errechnet aus der Verkaufszahl, den variablen und den festen Kosten den Kosten-Neutralpunkt. Ändert man die variablen Kosten, so kann man direkt die Wirkung auf diesen Punkt ablesen.



1 Variable Kosten
2 Verkäufe
3 Kosten-Neutralpunkt
4 Feste Kosten
5 Profit

Beispiel:

Verkäufe: 1000 Mark
Variable Kosten: 6000 Mark
Feste Kosten: 3000 Mark

Verändern Sie dann die variablen Kosten auf 5000 Mark und beachten Sie, wie sich der Kosten-Neutralpunkt verschiebt:

Die Formeln

$$\text{Variable Kostenrate} = \frac{\text{Variable Kosten}}{\text{Verkäufe}}$$

$$\text{Kosten-Neutralpunkt} = \frac{\text{Feste Kosten}}{1 - \text{Variable Kostenrate}}$$

Eingabe der Formeln als ESP

Variable Kostenrate	= VKstRt
Variable Kosten	= VarKstn
Verkäufe	= Verk
Kosten-Neutralpunkt	= KstnNPk
Feste Kosten	= FixKstn

```
=VKstRt=VarKstn/Verk:Kst
 nNPk=FixKstn/(1-VkstRt)
```

Anwendungssequenz

	Tastenfolge	Anzeige	Bemerkung
1	SHIFT CA	>	Initialisierung
2	= ENTER	=GrsScor=ZwSum =MZs =Rbt10% =VkstRt =ZensSum	Formel Menü
3	▶ ▶ ▶ ▶ ▶	=GrsScor=ZwSum =MZs =Rbt10% =VkstRt =ZensSum	Formel wählen
4	ENTER	Varkstn:Verk :FixKstn _	Abfrage variable Kosten
5	6000 ENTER	VarKstn:Verk :FixKstn 6000. :_	Abfrage Verkäufe
6	10000 ENTER	VarKstn:Verk :FixKstn 6000. :10000. :_	Abfrage Festkosten
7	3000 ENTER	Verk :FixKstn:VKstRt 10000. :3000. :0.6	Ausgabe Variable Rate
8	▶	FixKstn:VKstRt :KstnNPk 3000. :0.6 :7500	Ausgabe Neutralpunkt
9	◀ ◀ ◀ ◀	VarKstn:Verk :FixKstn 6000 :10000. :3000.	zurück zu variablen Kosten
10	5 ENTER	VarKstn:Verk :FixKstn 5000. :10000. :3000.	
11	ENTER	Verk :FixKstn:VKstRt 10000. :3000. :0.5	Ausgabe neue Rate
12	▶	FixKstn:VKstRt :KstnNPk 3000. :0.5 :6000.	Ausgabe neuer Neutralpunkt

SCHULNOTEN-SCHNITT

Den Schulnoten-Schnitt auf dem laufenden zu halten, ist nicht immer einfach. Die Aktualisierung nach jeder Prüfung ist eine nützliche Sache, um den Überblick zu behalten. Das folgende Beispiel-Programm berechnet die Notensumme und den Notenschnitt aus fünf Fächern.

Beispiel:

Name	Kunst	Engl	Math	Physik	Sport	Summe	Schnitt
Müller	1	2	1	2	3	9	1.8
Schmidt	3	1	1	1	3	9	1.8
Meier	3	2	3	2	2	12	2.4

Die Formeln

```
Summe = Kunsterziehung + Englisch + Mathematik +
        Physik + Sport

Notenschnitt = Summe/5
```

Eingabe der Formel als ESP:

```
=ZensSum=kunst+engl+math
 e+physik+sport:SCHNITT=
 ZenSum/5
```

Anwendungssequenz

	Tastenfolge	Anzeige	Bemerkung
1	SHIFT CA	>	Initialisierung
2	= ENTER	=GrsScor=ZwSum =MZs =Rbt10% =VkstRt ⌸ZensSum	Formel Menü
3	▶▶▶ ▶▶▶	=GrsScor=ZwSum =MZs =Rbt10% =VkstRt =ZensSum	Formel wählen
4	ENTER	kunst :engl :mathe _	Abfrage Kunst
5	1 ENTER	kunst :engl :mathe 1. :_	Abfrage Englisch
6	2 ENTER	kunst :engl :mathe 1. :2. :_	Abfrage Mathematik
7	1 ENTER	engl :mathe :physik 2. :1. :_	Abfrage Physik
8	2 ENTER	mathe :physik :sport 1. :2. :_	Abfrage Sport
9	3 ENTER	physik :sport ZensSum 2. :3. 9.	Ausgabe Summe
10	▶	sport :ZensSum:SCHNITT 3. :9. :1.8	Ausgabe Schnitt
11	ENTER	kunst :engl :mathe _	Wiederholung

BASIC-ANWENDER-PROGRAMME

NULLSTELLEN EINER FUNKTION

ÜBERBLICK

Die Nullstellen einer Funktion zu ermitteln, ist im allgemeinen sehr zeitraubend. Nachstehend ist ein Verfahren zur Nullstellenberechnung nach Newton ausgeführt. Bei dem Verfahren nach Newton variiert der Anfangspunkt automatisch gemäß dem festgelegten Intervall.

INHALT (Formel)

$$X_{n+1} = X_n - \frac{f(X_n)}{f'(X_n)}$$

Beträgt der absolute Wert in der Differenz zwischen X_n und X_{n+1} weniger als 10^{-8}, wird X_n als Nullstelle angezeigt.

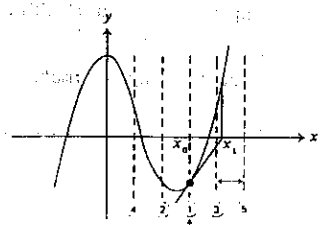
Der Differentialwert f'(X) wird wie folgt definiert:

$$f'(X) = \frac{f(X+h) - f(X)}{h} \qquad \text{(h: Minutenwert)}$$

Um 10^{-8} abzuwandeln, verändert man E-8 in Zeile 340.

ANLEITUNG ZUM BETRIEB

Eingabe: Anfangswert
Minutenwert
Intervall



Ausgabe: Nullstelle (Die ENTER -Taste drücken, um die Nullstelle im nächsten Intervall zu ermitteln.)

BEISPIEL

$x^3-2x^2-x+2 = 0$ (Nullstellen = -1, 1, 2)

Anfangspunkt = 0

Minutenwert = 10^{-4}

Intervall = 0,5

Die Funktion wird als Unterprogramm ab Zeile 500 geschrieben.

Dazu wird im PRO-Modus folgende Unterroutine eingegeben:

```
500 B = ((X-2)*X-1)*X+2    [ENTER]
510  RETURN [ENTER]
```

TASTENBETÄTIGUNG

Schritt-nr.	Eingabe	Anzeige	Bemerkungen
1	[DEF] A	ANFANGSWERT=_	
2	0 [ENTER]	MINUTE=_	
3	0.0001 [ENTER]	INTERVALL=_	
4	0.5 [ENTER]	ERGEBNIS=2.	
5	[ENTER]	ERGEBNIS=1.	Zum Auffinden der nächsten Nullstelle nochmal die [ENTER]-Taste drücken.
6	[ENTER]	ERGEBNIS=-1.	
7	[ENTER]	ERGEBNIS=1.	
8	[ENTER]	ERGEBNIS=-1.	
9	[ENTER]	ERGEBNIS=-1.	
10	[ENTER]	ERGEBNIS=-1.	
11	[ENTER]	ERGEBNIS=2.	

PROGRAMMLISTING

```
 10:"A": INPUT "ANFANGSW
   ERT=";V
 20:INPUT "MINUTE=";A
 30:INPUT "INTERVALL=";W
 40:G=V:F=V:Z=0
 50:IF Z=0 GOTO 70
 60:G=G-W:C=G: GOTO 80
 70:C=G:Z=1
 80:GOSUB 300
 90:F=F+W:C=F
100:GOSUB 300
110:GOTO 50
120:END
300:X=C: GOSUB 500
310:Y=B:X=A+C
320:GOSUB 500
330:D=C:C=D-A*Y/(B-Y)
340:IF ABS (D-C)>=1E-8
    GOTO 300
350:BEEP 3:PRINT "ERGEB
    NIS=";C
360:RETURN
500:B=((X-2)*X-1)*X+2
510:RETURN
520:END
```

SPEICHERINHALT

A	MINUTENWERT
B	f(X)
C	X0
D	f(x+h)
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	Anfangswert
W	Intervall
X	X
Y	f(x)
Z	Anfangsmarkierung

MITTELWERT, VARIANZ UND STANDARDABWEICHUNG

ÜBERBLICK

Nach Eingabe der Daten wird die Gesamtsumme, der Mittelwert, die Varianz und die Standardabweichung berechnet.
Korrektur der Eingabedaten als auch Daten mit Gewichtung möglich.

INHALT

Gesamtsumme $\Sigma x_i \cdot f_i$

Standardabweichung $\sigma = \sqrt{\sigma^2}$

Mittelwert $\bar{x} = \frac{\Sigma x_i \cdot f_i}{\Sigma f_i}$

Varianz $\sigma^2 = \frac{\Sigma (x_i - \bar{x}) f_i}{\Sigma f_i - 1}$

ANLEITUNG ZUM BETRIEB

1. DEF A : Wahl, ob Daten mit oder ohne Gewichtung verarbeitet werden sollen, Eingabe der Daten.
2. DEF B : Auffinden von Korrekturpositionen in den Daten.
   DEF C : Korrektur der Daten.
3. DEF D : Berechnung der Gesamtsumme, Mittelwert, Varianz und Standardabweichung.

BEISPIEL:

xi	14.1	14.2	14.3	14.4	14.5
fi	8	19	23	15	10

(Daten mit Gewichtung)

TASTENBETÄTIGUNG

Schritt-nr.	Eingabe	Anzeige	Bemerkungen
1	DEF A	ANZAHL DER DATEN=_	Eingabe der Anzahl der Daten
2	5 ENTER	ANZAHL DER DATEN=5_ GEWICHTUNG=1/KEINE=2?_	Wahl, ob Daten mit oder ohne Gewichtung
3	1 ENTER	X(1)=?	
4	14.1 ENTER	X(1)=14.1 F(1)=?	
5	8 ENTER	X(1)=14.1 X(2)=?	
	. . .	. . .	
12	14.5 ENTER	X(5)=14.5 F(5)=?	
13	10 ENTER	>	Ende des Programmteils

TASTENBETÄTIGUNG

Schritt nr.	Eingabe	Anzeige	Bemerkungen
1	DEF B	X(1)=14.1 F(1)=8	
2	ENTER	X(2)=14.1 F(2)=19	DEF C starten, um fehlerhafte Eingabe zu korrigieren.
3	DEF C	X(2)=14.1 X(2)=?	Eingabe des neuen Wertes
4	14.2 ENTER	F(2)=19 F(2)=?	
5	18 ENTER	X(2)=14.2 F(2)=18	
	. . .	. . .	
1	DEF D	GESAMTSUMME=1072.5 MITTELWERT=14.3	Anzeige der Gesamtsumme, Anzeige des Mittelwertes
2	ENTER	VARIANZ=1.432432432E-02	Anzeige der Varianz
3	ENTER	STANDARDABWEICHUNG= 1.196842693E-01	Anzeige der Standardabweichung
4	ENTER	>	Programmende

PROGRAMMLISTING

```
 10:"A": CLEAR : WAIT 0:
    CLS
 20:INPUT "ANZAHL DER DA
    TEN=";P
 30:CURSOR 24: INPUT "GE
    WICHTUNG=1/KEINE=2?"
    ;A
 35:CLS
 40:IF A=2 DIM X(P-1):
    GOTO 70
 50:IF A=1 DIM X(P-1),F(
    P-1): GOTO 70
 60:GOTO 30
 70:FOR I=0 TO P-1
 80:C$="X("+STR$ (I+1)+
    ")="
 85:PRINT C$;: INPUT X(I
    )
100:IF A=2 GOTO 141
110:CLS : PRINT C$;X(I)
120:B$="F("+ STR$ (I+1)+
    ")="
130:CURSOR 24: PRINT B$;
    : INPUT F(I)
141:CLS : PRINT "X(";
    STR$ (I+1);")=";X(I)
    : CURSOR 24
150:NEXT I: END
200:"B": WAIT 0:I=0
205:IF A=2 WAIT
210:B$="X("+ STR$ (I+1)+
    ")=":J=1: PRINT B$;X
    (I): CURSOR 24
230:IF A=1 LET C$="F("+
    STR$ (I+1)+")=":
    WAIT : PRINT C$;F(I)
    :J=2: GOTO 240
240:I=I+1
250:IF I=P END
255:CLS : WAIT 0: GOTO 2
    05
260:"C": WAIT 0: PRINT B
    $;X(I): CURSOR 24:
    PRINT B$;: INPUT X(I
    )
265:IF A=2 GOTO 250
270:CLS : PRINT C$;F(I):
    CURSOR 24: PRINTC$;
    : INPUT F(I)
280:GOTO 250
290:IF J=1 GOTO 230
300:"D":N=0:T=0:S=0: FOR
    I=0 TO P-1:X=X(I)
305:F=1: IF A=1 LET F=F(
    I)
310:N=N+F:T=T+F*X:S=S+F*
    X*X: NEXT I
400:WAIT 0:X=T/N:Q=(S-N*
    X*X)/(N-1):S=√Q:
    BEEP 3
405:CLS : PRINT "GESAMTS
    UMME=";T
406:CURSOR 24: WAIT:
    PRINT "MITTELWERT=";
    X
407:CLS: WAIT : BEEP 1:
    PRINT "MITTELWERT=";
    X
408:CLS : WAIT 0:  BEEP 1
    : PRINT "STANDARDABW
    EICHUNG=": CURSOR 33
    : WAIT : PRINT S
410:END
```

SPEICHERINHALT

A	
B	
C	
D	
E	
F	
G	
H	
I	
J	Marke
K	
L	
M	
N	
O	
P	Anzahl der Daten
Q	Varianz
R	
S	Standardabweichung
T	Gesamtsumme
U	
V	
W	
X	Mittelwert
Y	
Z	
X(P-1)	Daten
F(P-1)	Daten

SCHNITT ZWISCHEN KREISEN UND GERADEN

ÜBERBLICK

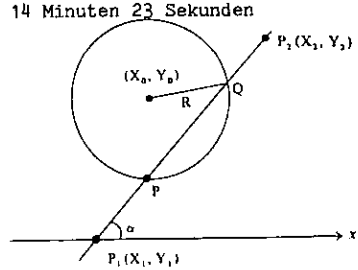
Es werden die Schnittpunkte zwischen Kreisen und Geraden in einer X-Y-Ebene bestimmt.

INHALT (Formeln)

Die Schnittpunkte zwischen einem Kreis und einer Geraden seien P und Q.

Hinweis: Die Winkel müssen in Grad, Minuten und Sekunden folgendermaßen eingegeben werden.

123.1423 = 123 Grad 14 Minuten 23 Sekunden



ANLEITUNG ZUM BETRIEB

1. Wird die Gerade über 2 Punkte bestimmt, so wird das Programm über DEF A gestartet.

   Wird die Gerade durch einen Punkt und einen Winkel bestimmt, starten Sie die Programmausführung über DEF B .

2. Nach der Dateneingabe werden die Ergebnisse ausgedruckt.

BEISPIEL

$X_1 = -50$

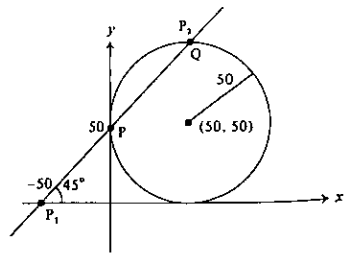
$Y_1 = 0$

$X_2 = 50$ $X_P = 0$

$Y_2 = 100$ $Y_P = 50$

$X_0 = 50$ $X_Q = 50$

$Y_0 = 50$ $Y_Q = 100$



$R = 50$

$\alpha = 45°$

Hinweis: Die Koordinatenwerte sind bis auf 5 Dezimalstellen genau.

TASTENBETÄTIGUNG (2 Punkte der Geraden sind bekannt)

Schritt-nr.	Eingabe		Anzeige	Bemerkungen
1	DEF	A	X0= ? Y0= R=	
2	50	ENTER	X0=50_ Y0= R=	
3	50	ENTER	X0=50_ Y0=50_ R= ?	
4	50	ENTER	X1= ? Y1=	
5	-50	ENTER	X1=-50_ Y1=?	
6	0	ENTER	X1=-50_ Y1=0_ X2=? Y2=	
7	50	ENTER	X1=-50_ Y1=0 X2=50_ Y2=?	
8	100	ENTER	P: 0.0000 49.9999	(XP,YP)
9		ENTER	P: 0.0000 49.9999 Q: 50.0000 100.0000	(XQ;YQ)

TASTENBETÄTIGUNG (1 Punkt und 1 Winkel sind bekannt)

Schritt nr.	Eingabe	Anzeige	Bemerkungen
1	DEF B	X0= ? Y0= R=	
2	50 ENTER	X0=50_ Y0=? R=	
3	50 ENTER	X0=50_ Y0=50_ R= ?	
4	50 ENTER	X1= ? Y1=	
5	-50 ENTER	X1=-50_ Y1=?	
6	0 ENTER	X1=-50_ Y1=0_ A= ?	
7	45 ENTER	P: 0.0000 49.9999	(XP,YP)
8	ENTER	P: 0.0000 49.9999 Q: 50.0000 100.0000	(XQ;YQ)
9	ENTER	>	

PROGRAMMLISTING

```
 10:"A":J=0: GOTO 25
 20:"B":J=1
 25:WAIT 0: PRINT "X0= "
    ,"Y0= ","R= "
 29:DEGREE
 30:CURSOR 4: INPUT A:
    CURSOR 16: INPUT B:
    CURSOR 27: INPUT C
 35:CLS : PRINT "X1= ","
    Y1= "
 40:CURSOR 4: INPUT D:
    CURSOR 16: INPUT E
 50:IF J<>0 CURSOR 24:
    PRINT "A= ": CURSOR
    27: INPUT H:H= DEG H
    : GOTO 90
 55:CURSOR 24: PRINT "X2
    = ": CURSOR 36:
    PRINT "Y2= "
 60:CURSOR 28: INPUT F:
    CURSOR 40: INPUT G:
 70:X=F-D:Y=G-E: GOSUB 5
    00
 80:H=X
 90:X=A-D:Y=B-E: GOSUB 5
    00
100:K=W* SIN (X-H)
110:L= ACS (K/C)
120:M=H-90-L:N=H-90+L
130:GOSUB 600
135:CLS : WAIT
140:PRINT USING "=======.
    ====";"P;":O;P
150:M=N: GOSUB 600
160:CURSOR 24: PRINT "Q:
    ";O;P
170:END
500:W= (X*X+Y*Y)
510:X= ACS (X/W): IF Y<0
    LET X=360-X
520:RETURN
600:O=A+C* COS M:P=B+C*
    SIN M: RETURN
610:END
```

SPEICHERINHALT

A	X0
B	Y0
C	R
D	X1
E	Y1
F	X2
G	Y2
H	
I	
J	
K	h
L	α
M	QP
N	QQ
O	XP, XQ
P	YP, YQ
Q	
R	
S	
T	
U	
V	
W	L
X	$\Delta X, \theta$
Y	ΔY
Z	

BERECHNUNG DER ANZAHL VON TAGEN ZWISCHEN ZWEI DATEN

ÜBERBLICK

Wieviele Tage sind seit Ihrer Geburt vergangen? Dieses Programm beantwortet solche Fragen. Nach Eingabe eines bestimmten Datums gibt das Programm die Anzahl der seitdem verstrichenen Tage an.

ANLEITUNG ZUM BETRIEB

Das Programm wird durch Drücken von [DEF] [A] initiiert.

Eingabe

[DEF] [A]

STARTJAHR	[ENTER]
MONAT	[ENTER]
TAG	[ENTER]
ZIELJAHR	[ENTER]
MONAT	[ENTER]
TAG	[ENTER]

Das Programm wird durch Eingabe von [DEF] [Z] anstelle der Jahreszahl beendet.

Beispiel

vom 5.10.1976

bis zum 4.06.1983 sind 2068 Tage
bis zum 1.01.1985 sind 3010 Tage

TASTENBETÄTIGUNG

Schritt nr.	Eingabe	Anzeige	Bemerkungen
1	DEF A	STJAHR? MON. TAG	
2	1976 ENTER	STJAHR1976_ MON.? TAG	Eingabe Ausgangs-datums: 5.10.1976
3	10 ENTER	STJAHR1976_ MON.10_ TAG?	
4	5 ENTER	STJAHR1976_ MON.10_ TAG5_ ZIJAHR? MON. TAG	
5	1982 ENTER	STJAHR1976_ MON.10_ TAG5_ ZIJAHR1982_ MON.? TAG	Eingabe Zieldatum: 4.6.1982
6	6 ENTER	STJAHR1976_ MON.10_ TAG5_ ZIJAHR1982_ MON.6_ TAG?	
7	4 ENTER	STJAHR1976_ MON.10_ TAG5_ ******2068.TAGE**********	
8	ENTER	STJAHR1976_ MON.10_ TAG5_ ZIJAHR? MON. TAG	
9	1985 ENTER	STJAHR1976_ MON.10_ TAG5_ ZIJAHR1985_ MON.? TAG?	Eingabe Zieldatum: 1.1.1985
10	1 ENTER	STJAHR1976_ MON.10_ TAG5_ ZIJAHR1985_ MON.1_ TAG?	
11	1 ENTER	STJAHR1976_ MON.10_ TAG5_ ******3010.TAGE**********	
12	ENTER	STJAHR1976_ MON.10_ TAG5_ ZIJAHR? MON. TAG	
13	DEF Z	>	Programmende

PROGRAMMLISTING

```
 10:"A": WAIT 0: USING :
    CLS
 20:PRINT "STJAHR      MO
    N.    TAG  "
 30:CURSOR 6: INPUT R
 40:CURSOR 15: INPUT S
 50:CURSOR 21: INPUT T
 60:H=R
 70:CURSOR 24: PRINT "ZI
    JAHR      MON.    TAG
      "
 80:CURSOR 30: INPUT F
 90:CURSOR 39: INPUT V
100:CURSOR 45: INPUT W
110:H=R
120:G=S:I=T
130:GOSUB 500
140:J=I:H=F
150:G=V:I=W
160:GOSUB 500
170:X=I-J
180:CURSOR 24: PRINT "**
    ****";X;"TAGE*******
    ****"
200:WAIT : PRINT
210:WAIT 0: GOTO 70
500:IF G-3>=0 LET G=G+1:
    GOTO 520
510:G=G+13:H=H-1
520:I= INT (365.25*H)+
    INT (30.6*G)+I
530:I=I- INT (H/100)+
    INT (H/400)-306-122:
    RETURN
600:"Z": END
```

SPEICHERINHALT

A	
B	
C	
D	
E	
F	JAHR (nach Berechnung)
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	Startjahr
S	Startmonat
T	Starttag
U	
V	Zielmonat
W	Zieltag
X	Anzahl der Tage
Y	
Z	

SCHREIBÜBUNGEN

ÜBERBLICK

Schnelle Tastenbetätigung bringt erhebliche Zeitersparnis. Können Sie die Tastatur schnell und fehlerfrei bedienen? Dieses Programm ist eine Übung, mit der Sie Ihr Tempo beim Eintasten erhöhen. Das Ergebnis ist eine schnelle, effektivere Programmeingabe ins Gerät.

INHALT

Die Anzahl der Buchstaben (3 bis 6) wird durch eine zufallszahlengenerierende Funktion festgelegt. Mit der gleichen Funktion wird der Zeichen-String (A bis Z) ausgewählt.

ANLEITUNG ZUM BETRIEB

Wenn der Summer ertönt, wird eine Schreibübung von 3 bis 6 Buchstaben angezeigt. Mit der Tastatur gibt man die gleichen Buchstaben innerhalb der vorgegebenen Zeit ein. Bei korrekter Eingabe erhält man 10 Punkte, und 5 Punkte, wenn mehr als 50 % korrekt ist. Falls die vorgegebene Zeit überschritten wird, erscheint eine weitere Übung in der Anzeige.

Die festgelegte Zeit richtet sich nach der Zahl der angezeigten Buchstaben und nach den Schwierigkeitsgraden (1, 2, 3). Bei Grad 1 ist die Zeit am kürzesten und bei Grad 3 am längsten. Für jeden Grad gibt es 10 Schreibübungen. Es gilt, möglichst 100 Punkte zu erreichen.

TASTENBETÄTIGUNG

Schritt-nr.	Eingabe	Anzeige	Bemerkungen
1	DEF Z	SCHREIBUEBUNG GRAD (1, 2, 3)?_	Eingabe des Schwierigkeits-grades
2	1 ENTER	SCHREIBUEBUNG HOECHSTE PUNKTZAHL=0	
3	A	SCHREIBUEBUNG ACFG	
4	C	SCHREIBUEBUNG ACFG A	
		SCHREIBUEBUNG ACFG AC	
. . .	. . .	. . .	
		IHRE PUNKTZAHL=80. HOECHSTE PUNKTZAHL=90.	Nach 10 Spiel-durchläufen wird die Punktzahl angezeigt.
1	ENTER	SCHREIBUEBUNG GRAD (1, 2, 3)?_	
	BREAK		Abbruch des Programms

PROGRAMMLISTING

```
10:"Z": CLEAR : DIM B$(
   6),Cs(5): RANDOM
13:WAIT 0: CLS: CURSOR
   7: PRINT "SCHREIBUEB
   UNG": CURSOR 29
15:INPUT "GRAD(1,2,3)?"
   ;L: WAIT 0
17:IF (L=1)+(L=2)+(L=3)
   <>1 THEN 13
20:"A": WAIT 0:P=0: CLS
   : CURSOR 7: PRINT "S
   CHREIBUEBUNG":
   CURSOR 25: PAUSE "HO
   ECHSTE PUNKTZAHL=";X
30:FOR S=1 TO 10
40:B= RND 4+2:Y$="":R=
   INT (B/2)
50:FOR C=0 TO B-1
60:D= RND 26:B$(C)=
   CHR$ (D+&40):Y$=Y$+
   CHR$ (D+&40): NEXT C
70:BEEP 3:E=0
75:CLS : PRINT "SCHREIB
   UEBUNG"; RIGHT$ (" "
   + STR$ (S),2);":";Y$
   :CURSOR 36
80:FOR W=1 TO B*10/L*2:
   IF E=B LET W=B*20/L*
   2: GOTO 100
85:LET C$(E)= INKEY$ :
   IF C$(E)="" THEN 100
87:A$=A$+C$(E): PRINT C
   $(E);
90:E=E+1
100:NEXT W:Q=0
110:FOR W=0 TO B-1: IF B
   $(W)=C$(W) LET Q=Q+1
120:NEXT W: IF Q<=R THEN
   150
130:IF Q=B LET P=P+10:
   GOTO 150
140:P=P+5
150:NEXT S: BEEP 3:CLS
   : WAIT 100: PRINT "I
   HRE PUNKTZAHL=";P
160:IF P>X LET X=P
165:WAIT : CURSOR 25:
   PRINT "HOECHSTE PUNK
   TZAHL=";X
170:GOTO 13
```

SPEICHERINHALT

A$	
B	
C	Schleifenvariable
D	
E	
F	
G	
H	
I	
J	
K	
L	Schwierigkeitsgrad
M	
N	
O	
P	Punktzahl
Q	
R	
S	Schleifenvariable
T	
U	
V	
W	Schleifenvariable
X	Höchste Punktzahl
Y$	
Z	
B$(5)	
C$(5)	

MONDLANDUNG

ÜBERBLICK

In diesem Spiel geht es darum, eine Rakete, die nur einen begrenzten Treibstoffvorrat hat, möglichst sanft zu landen. Die Rakete ist im freien Fall. Die Maschine wird benutzt, um den freien Fall zu bremsen. Findet die Zündung zu früh statt oder wird zuviel Treibstoff verbraucht, wird die Rakete zurück ins All geworfen und verglüht.

Ist der Treibstoff verbraucht, schlägt sie auf dem Planeten auf und explodiert.

Das Ziel ist, die Rakete so sanft wie möglich zu landen, indem man die Maschine kontrolliert und den Treibstoffverbrauch beobachtet.

INHALT (Formeln)

Die Schwerkraft bei 5m/(Zeiteinheit)^2.

Werden 5 Treibstoffeinheiten pro Zeiteinheit verbraucht, wird die Schwerkraft ausgeglichen.

$H = H_0 + V_0 t + \frac{1}{2} at^2$

$V = V_0 + at$

$V^2 = V_0^2 + 2aH$

$H_0 = 500,\ V_0 = -50,\ F_0 = 200$

H : Höhe
V : Geschwindigkeit
a : Beschleunigung
t : Zeit

H_0 : Anfangshöhe
V_0 : Anfangsgeschwindigkeit
F_0 : Treibstoffmenge
F : Verbrauchter Treibstoff

Die Anfangshöhe, Treibstoffmenge und Zeit sind in Zeile 30 durch die DATA-Anweisung gespeichert. Die Mengen können durch Umschreiben der Programmzeile geändert werden.

ANLEITUNG ZUM BETRIEB

1. Das Programm wird über DEF A gestartet. Über Ø-9 wird die vor Landung der Rakete benötigte Treibstoffmenge ausgeglichen.

TASTENBETÄTIGUNG

Schritt-nr.	Eingabe	Anzeige	Bemerkungen
1	DEF A	***START***	
			Tasten 0 - 9 geben Treibstoffmenge/ Zeiteinheit an.
		H:500 S:-50 F:200 C=0	
2	9	H:452 S:-46 F:191 C=9	
3 . . .	9 . . .	Wiederholung . . .	
	falls erfolgreich	ERFOLG !!	
		VERBLEIB: F=15	
	falls erfolglos	ENDE !!	
		NEUES SPIEL (J/N)?	Eingabe, ob ein neues Spiel gewünscht wird.
	J		Neues Spiel
	N	>	Ende

PROGRAMMLISTING

```
 10:"A": WAIT 50: CLEAR
    : USING :S=-50:A=0:D
    $=""
 20:BEEP 3: PRINT "  ***
     START ***"
 30:DATA "ZEIT=",50,"TRE
    IBSTOFF=",200,"HOEHE
    =",500
 40:RESTORE
 50:READ B$,W,B$,F,B$,H
 60:WAIT W
 70:PRINT USING "≡≡≡≡";"
    H:";H;" S:";S;" F:";
    F;" C:"; STR$ C
 80:IF F<=0 GOTO 170
 90:BEEP 1:D$= INKEY$
100:IF D$="" LET C=A:
    GOTO 130
110:C= VAL D$
120:A=C
130:IF C>F LET C=F
140:F=F-C:X=C-5:H=H+S+X/
    2:S=S+X
150:IF H>0 GOTO 70
160:IF ( ABS H<5)+( ABS
    S<5)=2 BEEP 5: PRINT
    "ERFOLG !!"; GOTO 18
    0
170:BEEP 3: PRINT "ENDE
    !!": GOTO 190
180:WAIT 150: PRINT
    USING "≡≡≡≡";"VERBLE
    IB :F=";F
190:WAIT 50: PRINT "NEUE
    S SP IEL (J/N)?":Z$=
    INKEY$
200:IF (Z$="J")+(Z$="N")
    <>1 GOTO 190
210:IF Z$="J" GOTO 10
220:END
```

SPEICHERINHALT

A	
B	
C	Verbrauchter Treibstoff
D$	Verbrauchter Treibstoff
E	
F	Anfangstreibstoffmenge/ Treibstoffmenge
G	
H	Anfangshöhe/Höhe
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	Geschwindigkeit
T	
U	
V	
W	Zeit
X	
Y	
Z	

GEDÄCHTNISTEST

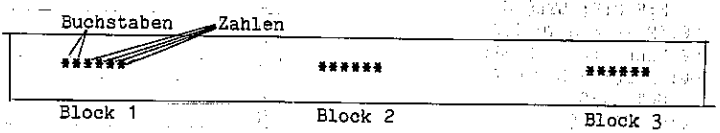
ÜBERBLICK

3 Zeilen mit 18 Zahlen werden ca. 5 sec. lang angezeigt.

Sie müssen, nachdem die Anzeige verschwunden ist, die gerade gesehenen Zeichen eingeben.

INHALT (Formeln)

Z. B. wird eine solche Zeichenfolge ca. 5 sec. lang angezeigt. Es handelt sich um 2 Buchstaben und 4 Zahlen.



* Die 3 Blocks müssen im Gedächtnis behalten und dann als Antwort eingegeben werden.

* Der Computer analysiert Ihre Antwort und wirft eine der 7 möglichen Auswertungen aus.

  Jeder Block ist in 2 Teile geteilt mit je 3 Zeichen. Sind alle Antworten richtig, ergibt es eine Punktzahl von 6.

Punkte	Ergebnis
0	Idiot
1	Schlecht
2	Mittelmäßig
3	Ok
4	Gut
5	Intelligent
6	Genial

TASTENBETÄTIGUNG

Schritt-nr.	Eingabe	Anzeige	Bemerkungen
1	DEF A	GEDAECHTNISTEST	Titel
2		GEDAECHTNISTEST AB1234 PK4398 VC7216	Anzeige Zeichenfolge (5 sec.)
3		ANTW.=_	Eingabe 1
4	AB1234 ENTER	ANTW.=_	Eingabe 2
5	PK4398 ENTER	ANTW.=_	Eingabe 4
6	VC7216 ENTER	AB1234 PK4398 VC7216	Angabe Zeichenfolge (1,5 sec.)
		AB1234 PK4398 VC7216 IDIOT SCHLECHT MITTELMAESSIG OK GUT! **INTELLIGENT**** ***GENIAL****	Anzeige Antwort Anzeige der Kategorie
		OK NEUES SPIEL (J/N)?_	Anzeige der Kategorie, Neues Spiel?
7	J oder N ENTER		Wenn J, GOTO Schritt 2
		>	Wenn N, END

PROGRAMMLISTING

```
 10:"A": CLS : USING :
    WAIT 0: CURSOR 4:
    PRINT "GEDAECHTNISTE
    ST": CLEAR
 20:DIM G$(6)*1,N$(10)*1
   ,V$(3)*3,X$(3)*6,Z$(
   3)*3,Y$(3)*6
 30:FOR I=1 TO 9:N$(I)=
   STR$ I: NEXT I:N$(10
   )="0"
 50:FOR I=1 TO 6
 60:J= RND 26:J=J+64
 70:G$(I)= CHR$ (J):
   NEXT I
 80:FOR I=1 TO 3
 90:Y$(I)=" "
100:FOR J=1 TO 3:K= RND
   9
110:Y$(I)=Y$(I)+N$(K):
   NEXT J
120:L= RND 9:J=(I-1)*2+1
130:A$(I)=G$(J)+G$(J+1)+
   N$(L)
140:H$=Y$(I):A$(I+3)=
   RIGHT$ (H$,3): NEXT
   I
150:CURSOR 25: WAIT 300:
   GOSUB 500: WAIT 0
160:FOR I=1 TO 3
170:CLS : INPUT "ANTWORT
   =";X$(I):X$(I)=
   LEFT$ (X$(I),6)
180:Z$(I)= LEFT$ (X$(I),
   3)
190:V$(I)= RIGHT$ (X$(I)
   ,3): NEXT I
200:CURSOR 1: GOSUB 500:
   CURSOR 25: GOSUB 520
210:N=0
220:FOR I=1 TO 3
230:IF A$(I)=Z$(I) LET N
   N=N+1
240:IF A$(I+3)=V$(I) LET
   N=N+1
250:NEXT I
260:N=N+1
265:WAIT 150
270:CLS : ON N GOTO 300,
   310,320,330,340,350,
   360
300:BEEP 1: PRINT "
       IDIOT": GOTO 370
310:BEEP 1: PRINT "
   SCHLECHT": GOTO 370
320:BEEP 2: PRINT "    M
   ITTELMAESSIG": GOTO 370
330:BEEP 2: PRINT "
         OK": GOTO 370
340:BEEP 3: PRINT "*****
   ****GUT********!:
   GOTO 370
350:BEEP 4: PRINT "*****
   *INTELLIGENT******":
   GOTO 370
360:BEEP 5: PRINT "*****
   ****GENIAL*********"
370:W$="": BEEP 1:
   CURSOR 25: INPUT "NE
   UES SPIEL (J/N)";W$
380:IF W$="N" THEN 600
390:IF W$="J" THEN 10
395:WAIT 0: GOTO 270
500:BEEP 2: PRINT A$(1);
   A$(4);"  ";A$(2);A$(
   5);"  ";A$(3);A$(6)
510:RETURN
520:BEEP 1: PRINT USING
   "&&&&&&";X$(1);
   USING;"   "; USING "
   &&&&&&";X$(2);
525:WAIT 80: PRINT USING
   ;"  "; USING "&&&&&&
   ";X$(2)
530:USING : RETURN
600:END
```

SPEICHERINHALT

A$	
B$	
C$	3 Spalten BuchstabenDIM A$(6)
D$	
E$	
F$	
G	
H$	
I	Index
J	Zufallszahlengenerator
K	Zufallszahlengenerator
L	Zufallszahlengenerator
M	
N	Zähler
O	
P	
Q	
R	
S	
T	
U	
V	
W$	Eingabe neues Spiel
X	
Y	
Z	
G$(6)	Buchstaben (1~6)
N$(6)	Arbeit (1~6)
V$(3)	3 Spalten nach der Antwort
X$(3)	Arbeit (1~3)
Y$(3)	Arbeit (1~3)
Z$(3)	3 Spalten vor der Arbeit

KÄFERJAGD

ÜBERBLICK

Der Käfer bewegt sich durch einen Zufallszahlengenerator. Der Mann jagt den Käfer und tötet ihn.

Der Mann bewegt sich über die Tasten 4 ← 8 ↑ 2 ↓ → 6 (INKEY$).

Jedes Mal, wenn der Mann sich um einen Schritt bewegt, tut der Käfer das auch (manchmal bleibt der Käfer auch stehen). Die Ausgangsposition des Mannes ist (0,0).

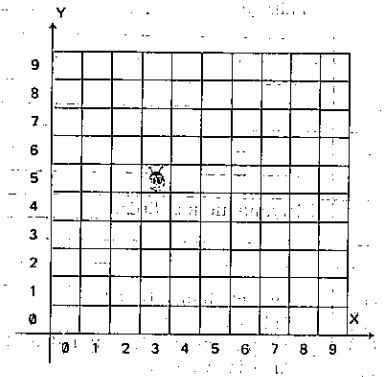
Die Ausgangsposition des Käfers wird über einen Zufallszahlengenerator gewählt. Hinweise werden angezeigt.

Der Abstand wird über die Gleichung ABS (X-A) + ABS (Y-B) berechnet und angezeigt. Die Ausgangsenergie ist 100. Sie nimmt bei jedem Schritt um 1 ab.

Jedes Mal, wenn ein Käfer getötet wird, steigt die Energie um 5, 10 oder 15 (der Betrag wird über einen Zufallszahlengenerator gewählt).

Die Punktzahl wird nach der Zahl der getöteten Käfer bei Energie = 0 berechnet.

Das Programm kann über RUN [ENTER] oder [DEF] [A] gestartet werden.



Position des Mannes (X,Y)

Position des Käfers (a,b)

Zur Anzeige: Kleinbuchstaben geben aktuelle Werte an.

```
(x,y)                    ABSTAND = 1                    E = e
```

Ausgangsposition (X-Koordinate/Y-Koordinate) | Hinweis (Abstand) | Verbleibende Energie

* Jedes Mal, wenn der Mann sich bewegt, ändert sich die Anzeige.

Der Käfer ist gefangen.

Pause

```
T R E F F E R !
B A N G !  B A N G !
PUNKTE t    ENERGIE e
```

BEEP:

* Hinweis: Wenn der Abstand 1 ist, ertönt ein akustische Signal dreimal, ist der Abstand 2, zweimal und ist der Abstand 3 einmal.

* Ist der Abstand größer als 3, ertönt kein Signal.

* Ist der Käfer gefangen, gibt der Computer 5 Signale.

TASTENBETÄTIGUNG

Schritt-nr.	Eingabe	Anzeige	Bemerkungen
1	DEF A	(0,0) ABSTAND=5 E=100	
	8	(0,1) ABSTAND=4 E=99	
	6	(1,1) ABSTAND=2 E=98	BEEP 2
	8	TREFFER!	BBEP 5
		BANG! BANG!	
		PUNKTE 1 ENERGIE 108	

PROGRAMMLISTING

```
 10:"A": RANDOM : WAIT 2
    50: PRINT "**KAEFERJ
    AGD**": BEEP 3
 20:X=0:Y=0:E=100:F=100:
    T=0:S=0
 30:A= RND 9:B= RND 9
 40:L= ABS (X-A)+ ABS(Y
    -B)
 50:IF X=A AND Y=B GOTO
    400
100:IF L=1 BEEP 3
110:IF L=2 BEEP 2
120:IF L=3 BEEP 1
130:WAIT 50: PRINT "(";
    STR$ (X);","; STR$ (
    Y);")ABSTAND="; STR$
    (L);"E="; STR$ (E)
150:S=S+1:E=F- INT (S/2)
153:IF E<=0 THEN 500
155:G$= INKEY$ : OF G$="
    " GOTO 130
157:BEEP 1
160:IF G$="2" LET Y=Y-1:
    GOTO 210
170:IF G$="4" LET X=X-1
    GOTO 210
180:IF G$="6" LET X=X+1:
    GOTO 210
190:IF G$="8" LET Y=Y+1:
    GOTO 210
200:GOTO 150
210:IF X<0 LET X=0: GOTO
    150
220:IF Y<0 LET Y=0: GOTO
    150
230:IF X>9 LET X=9: GOTO
    150
240:IF Y>9 LET Y=9: GOTO
    150
250:IF X=A AND Y=B GOTO
    400
260:E=F- INT (S/2)
270:IF E<=0 GOTO 500
280:R= RND 5
290:IF R=1 LET B=B-1:
    GOTO 340
300:IF R=2 LET A=A-1:
    GOTO 340
310:IF R=3 LET A=A+1:
    GOTO 340
320:IF R=4 LET B=B+1:
    GOTO 340
340:IF A<0 OR A>9 GOTO 3
    70
350:IF B<0 OR B>9 GOTO 3
    70
360:GOTO 40
370:BEEP 4: PAUSE "***VO
    RSICHT***": GOTO 30
400:PAUSE "TREFFER!"
410:BEEP 5
420:PAUSE "BANG! BANG!"
430:T=T+1:C= RND 3*5:F=F
    +C
435:E=F- INT (S/2)
440:WAIT 100: PRINT "PUN
    KTE ";T;"ENERGIE ";E
450:GOTO 30
500:WAIT : PRINT "PUNKTE
    "; STR$ (T);" *SPIE
    L AUS!!"
510:END
```

SPEICHERINHALT

A	X-Koordinate
B	Y-Koordinate
C	Energiezähler
D	
E	Verbleibende Energie
F	Energie
G$	Tasteneingabe 2, 4, 6, 8
H	
I	
J	
K	
L	Abstand Käfer/Mann
M	
N	
O	
P	
Q	
R	Größe der Bewegung des Käfers
S	Verstrichene Zeit
T	Punkte
U	
V	
W	
X	Position Mann X-Koordinate
Y	Position Mann Y-Koordinate
Z	

DOPPELTE ROTATION

ÜBERBLICK

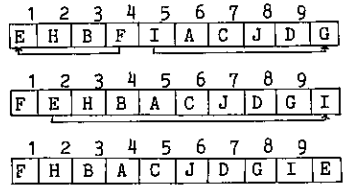
Ein Denkspiel, wobei willkürlich ausgelegte Buchstaben (A bis J) geordnet werden müssen. Schaffen Sie's beim ersten Mal? Vielleicht nicht. Versuchen Sie es trotzdem.

ANLEITUNG UND BETRIEB

DEF A "Doppelte Rotation" wird angezeigt. Daraufhin erscheinen die Buchstaben A, B, C ...J in willkürlicher Reihenfolge in der Anzeige. Mit der Eingabe von Fixpunkten (1 bis 9) werden die angezeigten Buchstaben "rotiert". Ihre Punktzahl wird durch die Häufigkeit der Tastenbetätigungen bestimmt. Je weniger es sind, desto besser.

BEISPIEL

Wenn z. B. der Fixpunkt 4 in diesen Zeichen-String eingegeben wird, werden die Buchstaben wie links im Bild dargestellt rotiert. Die nächste Darstellung erhält man durch Eintasten von Fixpunkt 1.



Versuchen Sie einmal auf diese Weise, das Ordnen möglichst effektiv zu gestalten.

TASTENBETÄTIGUNG

Schritt-nr.	Eingabe	Anzeige	Bemerkungen
1	DEF A ENTER	DOPPELTE ROTATION A∼J	Anzeige Zufalls-zeichenfolge
2	1 ∼ 9		Zahlen zwischen 1 und 9 werden gewählt und eingegeben.
		Eingabe Wiederholung	
		ABCDEFGHIJ	
		SPIEL AUS	
		PUNKTE 35	
		>	
1	DEF B	A ∼ J	Soll das neue Spiel mit der-selben Zu-fallszeichen-folge beginnen?
		Wie DEF A	

PROGRAMMLISTING

```
 10:"A": CLEAR : WAIT 50
    : RANDOM : DIM B$(4)
 20:PRINT "DOPPELTE ROTA
    TION"
 30:B$(0):"ABCDEFGHIJ"
 40:B$(1)=""
 50:A=0
 60:FOR I=1 TO 10
 70:R= RND 10
 80:S="^(R-1):B=S AND A
 90:IF B><0 GOTO 70
100:A=A OR S
110:B$(1)=B$(1)+ MID$ (B
    $(0),R,1): NEXT I
120:B$(2)=B$(1)
130:N=0
150:BEEP 1
170:D$="": PRINT B$(2):D
    $= INKEY$
180:C= VAL D$
190:IF C=0 GOTO 170
210:B$(3)= LEFT$ (B$(2),
    C)
220:B$(4)= RIGHT$ (B$(2)
    ,10-C)
240:IF C=1 GOTO 260
250:B$(3)= RIGHT$ (B$(3)
    ,1)+ LEFT$ (B$(3),C-
    1)
260:IF C=9 GOTO 280
270:B$(4) RIGHT$ (B$(4)
    ,9-C)+ LEFT$ (B$(4),
    1)
280:B$(2)=B$(3)+B$(4)
290:N=N+1
300:IF B$(2)><BS(0) GOTO
    150
310:BEEP 5: PRINT "SPIEL
    AUS"
320:WAIT 200: PRINT
    USING "====";"PUNKTE
    ";N
330:END
400:"B": WAIT 50; GOTO 1
    20
410:END
```

SPEICHERINHALT

A	
B	
C	
D$	Eingabe Taste
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	Punkte
O	
P	
Q	
R	Zufallszahlen
S	
T	
U	
V	
X	
Y	
Z	
B$(4)	Alphabet Folge

ANHANG A

FEHLERMELDUNGEN

Fehlernummer	Bedeutung
1	Syntax-Fehler Der PC 1260/1261 kann nicht verstehen, was Sie eingegeben haben. Prüfen Sie die Eingabe auf Dinge wie Semikolon am Ende eines PRINT-Statements, falsch geschriebene Befehle und fehlerhaften Gebrauch von Zeichen.
	3 */2
2	Rechenfehler Hier haben Sie vermutlich eines der drei folgenden Dinge getan: 1. Versucht, eine zu große Zahl zu benutzen; das Rechenergebnis ist größer als 9.999999999 E 99. 2. Versucht, durch Null zu teilen. (z. B. 5 / 0) 3. Versucht, eine unlogische Rechnung auszuführen. (z. B.: LN-30 oder ASN 1.5)
3	Unzulässige Funktion (DIMensionierungs-Fehler, Argument-Fehler) Die Feld-Variable existiert bereits. Sie haben versucht, ein Feld zu spezifizieren, ohne es zuvor zu dimensionieren. Der Index des Feldes übersteigt die im DIM-Statement vorgegebene Größe. (z. B.: DIM B(256)) Unzulässiges Funktionsargument, d. h. Sie haben versucht, Ihren Computer etwas tun zu lassen, wozu er nicht imstande ist. (z. B.: WAIT 66000)

4 Zu hohe Zeilennummer

Hier haben Sie vermutlich einen der folgenden Fehler gemacht:
1. Versucht, mit GOTO, GOSUB, RUN, LIST, THEN etc. eine nicht vorhandene Zeile anzusprechen.
2. Versucht, eine zu große Zeilennummer zu verwenden. Die höchstmögliche Zeilennummer ist 65279.

5 NEXT ohne FOR

Das Einrichten einer Subroutine übersteigt 10 levels.

Das Einrichten einer FOR-Schleife übersteigt 5 levels.

Die Befehle RETURN ohne GOSUB, NEXT ohne FOR, READ ohne DATA wurden benutzt.

Fassungsvermögen des Puffers überschritten.

6 Speicherkapazität überschritten

Diese Fehlermeldung erhalten Sie normalerweise, wenn Sie versuchen, ein Feld zu DIMensionieren, das zu groß für die Speicherkapazität ist. Sie kann aber auch auftreten,
- wenn ein Programm zu lang ist;
- wenn der Reservespeicher mehr als 48 Bytes enthält;
- wenn das Easy Simulation Programm zu umfangreich ist und die Kapazitätsgrenzen übersteigt.

7 PRINT USING Fehler

Das bedeutet, daß in einem USING-Statement eine unzulässige Format-Spezifikation enthalten ist.

8 I/O Device Fehler

Dieser Fehler kann nur auftreten, wenn Sie den zusätzlichen Drucker und/oder Cassettenrecorder an den PC 1260/1261 angeschlossen haben.

Die Meldung informiert Sie über ein Übermittlungsproblem zwischen dem Drucker und/oder Cassettenrecorder und dem PC 1260/1261.

9 Andere Fehler

Dieser Fehlercode wird angezeigt, wenn der Computer ein Problem hat, das mit den anderen Fehlercodes nicht zu erfassen ist. Eine der häufigsten Ursachen für das Auftreten dieser Meldung ist der Versuch, Daten einer Variablen unter einem bestimmten Namen anzusprechen (z. B. A$), während die Daten der Variablen unter dem anderen Namen (d. h. A) abgespeichert worden waren.

EINGABEFEHLER

Wenn Sie ein Programm ausführen, kann eine Fehlermeldung aufgrund eines Eingabefehlers auftreten. Beachten Sie in diesem Fall die folgenden Anmerkungen:

Beispiel:

Sie haben KPRINT anstelle von LPRINT eingegeben.

EINGABE	ANZEIGE
	10: K PRINT A$
►	10 K PRINT A$
L ENTER	10: L PRINT A$

Space

Wenn Sie auf diese Weise die Korrektur durchführen, versteht der Computer sie nicht als Befehl. In diesem Beispiel müssen Sie KPRINT löschen und dann LPRINT neu eingeben.

EINGABE	ANZEIGE
	10: K PRINT A$
► ►	10 K PRINT A$
SHIFT DEL SHIFT DEL	10 A$
SHIFT INS SHIFT INS	10 A$
LPRINT ENTER	10: LPRINT A$

Kein Space, da als Befehl erkannt.

Mit der Cursor-Taste können Sie den Befehl auf korrekte Eingabe überprüfen.

RICHTIGE EINGABE

```
    10: RADIAN
 ►  10  ▯ADIAN
 ►  10  RADIAN_
```

FALSCHE EINGABE

```
    10: RADAN
 ►  10  ▯ADAN
 ►  10  R▯DAN
 ►  10  RA▯AN
 ►  10  RAD▯N
```

ANHANG B

ASCII-CODE TABELLE

ASCII Zeichen	Entsprechender Code		
	Dual	Dezimal	Hexadezimal
Leerraum	00100000	32	20
!	00100001	33	21
"	00100010	34	22
=	00100011	35	23
$	00100100	36	24
%	00100101	37	25
&	00100110	38	26
'	00100111	39	27
(	00101000	40	28
)	00101001	41	29
*	00101010	42	2A
+	00101011	43	2B
,	00101100	44	2C
-	00101101	45	2D
.	00101110	46	2E
/	00101111	47	2F
0	00110000	48	30
1	00110001	49	31
2	00110010	50	32
3	00110011	51	33
4	00110100	52	34
5	00110101	53	35
6	00110110	54	36
7	00110111	55	37
8	00111000	56	38
9	00111001	57	39
:	00111010	58	3A
;	00111011	59	3B
<	00111100	60	3C
=	00111101	61	3D
>	00111110	62	3E
?	00111111	63	3F
@	01000000	64	40

ASCII Zeichen	Entsprechender Code		
	Dual	Dezimal	Hexadezimal
A	01000001	65	41
B	01000010	66	42
C	01000011	67	43
D	01000100	68	44
E	01000101	69	45
F	01000110	70	46
G	01000111	71	47
H	01001000	72	48
I	01001001	73	49
J	01001010	74	4A
K	01001011	75	4B
L	01001100	76	4C
M	01001101	77	4D
N	01001110	78	4E
O	01001111	79	4F
P	01010000	80	50
Q	01010001	81	51
R	01010010	82	52
S	01010011	83	53
T	01010100	84	54
U	01010101	85	55
V	01010110	86	56
W	01010111	87	57
X	01011000	88	58
Y	01011001	89	59
Z	01011010	90	5A
[	01011011	91	5B
]	01011101	93	5D
^	01011110	94	5E
_	01011111	95	5F
π	11111011	251	FB
√	11111100	252	FC

ANHANG C

FORMATIEREN DER DATENAUSGABE

Es ist manchmal wichtig oder nützlich, neben dem Inhalt ausgegebener Daten auch das Format zu kontrollieren. Der PC 1260/1261 bedient sich hierzu des Befehls USING. Mit Hilfe dieses Befehls können Sie spezifizieren:

- die Anzahl der Stellen
- die Position des Dezimalpunktes
- wissenschaftliche Notation
- Anzahl der Zeichen in einem String

Diese verschiedene Formate bestimmen Sie mit einer "Ausgabe Maske", die aus einer String-Konstanten oder auch einer String-Variablen bestehen kann.

```
10: USING "####"
20: M$ = "&&&&&"
30: USING M$
```

Wird der Befehl USING ohne Maske benutzt, werden damit alle Spezialformate aufgehoben.

```
40: USING
```

Der Befehl USING kann auch in einem PRINT-Statement benutzt werden.

```
50: PRINT USING M$;N
```

Wann immer USING benutzt wird, bewirkt es die Kontrolle aller ausgegebenen Daten, bis ein neuer USING-Befehl auftritt.

NUMERISCHE DATEN

Eine numerische USING-Maske darf nur benutzt werden, um numerische Werte, d. h. numerische Konstante oder numerische Variable zu kontrollieren. Wird eine String-Konstante oder -Variable angezeigt, während eine numerische USING-Maske wirksam ist, wird die Maske hierauf nicht angewendet werden.

Ein auszugebender Wert muß immer in den von der Maske vorgesehenen Freiraum passen. Es muß auch Raum für das Vorzeichen vorgesehen sein, selbst, wenn die Zahl immer positiv sein wird. Eine Maske, die vier Stellen zuläßt, kann also nur für dreistellige Zahlen verwendet werden.

BESTIMMUNG NUMERISCHER MASKEN

Die gewünschte Anzahl von Stellen wird mit Hilfe des "#"-Zeichens festgelegt. Die Anzeige oder der Ausdruck beinhalten immer soviele Zeichen, wie in der Maske vorgesehen sind. Dabei erscheint die Zahl rechts in diesem Bereich, die restlichen Stellen werden durch Leerschritte aufgefüllt. Bei positiven Zahlen wird sich daher auf der linken Seite der Anzeige immer mindestens ein Leerschritt befinden. Da der PC 1260/1261 nicht mehr als 10 Stellen erfassen kann, sollte die größte verwendete numerische Maske maximal 11 "#"-Zeichen enthalten.

Wenn die Gesamtstellenzahl des ganzzahligen Teils 11 Stellen übersteigt, wird dieser ganzzahlige Teil als 11 Stellen aufgefaßt.

Anmerkung: In allen Beispielen dieses Anhangs werden Sie am Anfang und am Ende eines Anzeigefeldes "|"-Zeichen finden, mit denen die Größe des Feldes anschaulich gemacht werden soll.

EINGABE	ANZEIGE
10: USING "#####"	(Bringen Sie den PC 1260/1261 in den RUN-Mode, geben Sie RUN ein und drücken Sie die [ENTER] -Taste.)
20: PRINT 25	| 25|
30: PRINT -350	|-350|
40: PRINT 1000	ERROR 7 in 40

Beachten Sie, daß die letzte Eingabe eine Fehlermeldung hervorruft, weil 5 Stellen (4 Zahlen und eine Stelle für das Vorzeichen) benötigt werden, die Maske aber nur 4 vorsieht.

BESTIMMUNG DES DEZIMALPUNKTES

Das Zeichen für den Dezimalpunkt (".") kann in einer Maske enthalten sein, um die gewünschte Position des Dezimalpunktes festzulegen. Wenn die Maske mehr Stellen bereitstellt, als für den anzuzeigenden Wert benötigt werden, werden die auf der rechten Seite übrigbleibenden Stellen mit Nullen aufgefüllt. Enthält der anzuzeigende Wert mehr Stellen, als die Maske vorsieht, werden diese abgeschnitten (nicht gerundet).

EINGABE	ANZEIGE
10: USING "#####.##"	
20: PRINT 25	| 25.00|
30: PRINT -350.5	|-350.50|
40: PRINT 2.547	| 2.54|

BESTIMMUNG DER WISSENSCHAFTLICHEN NOTATION

Das "^"-Zeichen kann in einer Maske enthalten sein, um so anzuzeigen, daß die Zahl in wissenschaftlicher Notation ausgegeben werden soll. Die Zeichen "#" und "." werden in der Maske benutzt, um den "charakteristischen" Teil der Zahl (d. h. den Teil, der links vom "E" steht) zu formatieren. Links vom Dezimalpunkt sollten immer 2 "#"-Zeichen stehen, um genug Raum für eine ganze Zahl und das Vorzeichen zu schaffen. Der Dezimalpunkt kann - muß aber nicht - vorgegeben werden.

Rechts vom Dezimalpunkt können bis zu 9 Stellen eingerichtet werden (mit Hilfe von "#"-Zeichen). Nach dem "charakteristischen" Teil wird das Exponentiationszeichen (E) angezeigt, gefolgt von der Stelle für das Vorzeichen und dem Exponenten (2stellig). Das kleinste von einer Maske vorzugebende Feld für wissenschaftliche Notation wäre demnach "##^", damit werden Zahlen in der Form "2 E 99" dargestellt. Das größtmögliche Feld für wissenschaftliche Notation wäre "##.#########^", das Zahlen wie "-1.234567890 E -12" darstellen kann.

EINGABE	ANZEIGE
10: USING "##.##^"	
20: PRINT 2	| 2.00 E 00|
30: PRINT -365.278	|-3.65 E 02|

BESTIMMUNG ALPHANUMERISCHER MASKEN

String-Konstanten und -Variable werden mit Hilfe des "&"-Zeichens ausgegeben. Jedes "&" gibt eine im vorgesehenen Feld anzuzeigende Stelle an. Der String wird auf der linken Seite dieses Feldes angezeigt. Ist der String kürzer als der dafür eingeräumte Raum, werden die rechts verbleibenden Stellen mit Leerschritten aufgefüllt. Ist der String länger als das Feld, wird er abgeschnitten:

EINGABE	ANZEIGE
10: USING "&&&&&&"	
20: PRINT "ABC"	|ABC |
30: PRINT "ABCDEFGHI"	|ABCDEF|

GEMISCHTE MASKEN

In den meisten Anwendungsfällen wird eine USING-Maske entweder alle notwendigen Zeichen zur Formatierung eines numerischen Ausdrucks oder zur Formatierung eines Strings beinhalten. Für bestimmte Zwecke können aber auch beide gemeinsam in einer USING-Maske enthalten sein. In solchen Fällen markiert jede Umschaltung von String-formatierenden Zeichen auf Zahlen-formatierende Zeichen (und umgekehrt) die Grenze für einen bestimmten Wert. So können mit einer Maske der Form "###.##&&&&&" zwei verschiedene Werte formatiert werden - ein numerischer Ausdruck, für den 5 Stellen vorgesehen sind, und ein alphanumerischer Ausdruck, für den 4 Stellen bereitgestellt wurden.

EINGABE	ANZEIGE
10: PRINT USING "==#=.##=&&";25;"CR"	| 25.00CR|
20: PRINT -5.789;"DB"	| -5.78DB|

Achtung: Wurde ein USING-Statement einmal spezifiziert, wirkt es sich auf alle nachfolgenden Daten aus, bis es aufgehoben oder durch einen anderen USING-Befehl ersetzt wird.

ANHANG D

BEWERTUNG VON AUSDRÜCKEN UND OPERATOREN-VORRANG

Wenn in den SHARP PC 1260/1261 ein komplexer Ausdruck eingegeben wird, bewertet er Teile dieses Ausdrucks in einer Reihenfolge, die durch die Vorrangstellung der einzelnen Teile bestimmt wird. Geben Sie den Ausdruck

```
100/5+45
```

ein, entweder als Rechenoperation oder als Teil eines Programms, so weiß der PC 1260/1261 nicht, ob Sie meinen:

$$\frac{100}{5 + 45} = 2 \quad \text{oder} \quad \frac{100}{5} + 45 = 65$$

Da der PC 1260/1261 eine Möglichkeit haben muß, zwischen diesen beiden Operationen zu entscheiden, bedient er sich seiner Regeln des Operatoren-Vorrangs. Da die Division eine höhere "Priorität" hat als die Addition (siehe unten), wird er entscheiden, daß zuerst die Division durchgeführt wird und anschließend die Addition, d. h. die zweite Möglichkeit wird ausgeführt und als Ergebnis 65 ausgegeben.

VORRANG VON OPERATOREN

Im BASIC-Modus werden Operatoren vom SHARP PC 1260/1261 gemäß den folgenden Prioritätsregeln verarbeitet (angefangen mit der höchsten Priorität):

1. Klammern
2. Variable und Pseudovariable
3. Funktionen
4. Exponentiation (^)
5. Vorzeichen (+, -)
6. Multiplikation und Division (*, /)
7. Addition und Subtraktion (+, -)
8. Verhältnis-Operatoren (<, <=, =, <>, >=, >)
9. Logische Operatoren (AND, OR)

Treten in einem Audruck zwei oder mehr Operatoren derselben Prioritätsstufe auf, wird der Ausdruck von links nach rechts verarbeitet. (Exponentiation wird von rechts nach links verarbeitet!) Beachten Sie, daß bei einer Operation A+B-C das Ergebnis dasselbe ist, ob Sie nun die Addition oder die Subtraktion zuerst ausführen.

Enthält ein Ausdruck ineinanderliegende Klammern, so wird die innerste Klammer zuerst bearbeitet, die äußerste zuletzt.

Für die Prioritätsstufen 3 und 4 gilt, daß die letzte Eingabe die höchste Priorität hat.

Z. B.:

$-2\uparrow 4 \longrightarrow -(2^4)$

$3\uparrow -2 \longrightarrow 3^{-2}$

BEISPIEL FÜR EINE BEWERTUNGSFOLGE

Wir gehen aus von dem Ausdruck:

```
((3+5-2)*6+2)/10↑LOG 100
```

Der PC 1260/1261 verarbeitet nun zuerst die innersten Klammern. Da '+' und "-" auf derselben Stufe stehen, wird von links nach rechts gerechnet, also die Addition zuerst ausgeführt.

```
((8-2)*6+2)/10↑LOG 100
```

Dann wird die Subtraktion durchgeführt:

```
((6)*6+2)/10↑LOG 100
```

Oder:

```
(6*6+2)/10↑LOG 100
```

In der nächsten Klammer wird zuerst die Multiplikation durchgeführt:

```
(36+2)/10↑LOG 100
```

Dann folgt die Addition:

```
(38)/10↑LOG 100
```

Oder:

```
38/10↑LOG 100
```

Nachdem nun die Klammern aufgelöst sind, hat die LOG-Funktion höchste Priorität.

```
38/10↑2
```

Als nächstes folgt die Exponentiation:

```
38/100
```

Und zuletzt wird die Division ausgeführt:

0.38

Dies ist der Wert des Ausdrucks.

ANHANG E

TASTENFUNKTIONEN

ON
BRK

(ON) wird benutzt, um den PC 1260/1261 anzuschalten, wenn er sich automatisch abgeschaltet hatte.

(BREAK)
Das Betätigen dieser Taste während eines Programmablaufs bewirkt eine Unterbrechung der Programmausführung.

Bei manuellen Operationen, Ein-/Ausgabe-Befehlen wie BEEP, CLOAD etc. wird mit Betätigung dieser Taste die Befehlsausführung unterbrochen.

SHIFT

Die gelbe Taste mit der Aufschrift "SHIFT" muß benutzt werden, um Doppelfunktionen anzusprechen (die jeweils in brauner Schrift über den entsprechenden Tasten stehenden Zeichen). Z. B. wird mit SHIFT und ? das "?" angesprochen.

CL

Wird benutzt, um Eingabe und Anzeige zu löschen, weiterhin hebt diese Taste eventuelle Blockaden durch Fehler auf.

SHIFT CA

Löscht nicht nur den Anzeigeinhalt, sondern initialisiert darüber hinaus den Computer, d. h.

- hebt den WAIT-Timer auf;
- löscht das Anzeigeformat (USING-Format);
- hebt den TRON-Befehl auf (TROFF);
- hebt PRINT=LPRINT auf;
- hebt Blockade durch Fehler auf.

0 bis 9

Zifferntasten

.

Dezimalpunkt

- wird benutzt, um eine Abkürzung eines Befehls, eines Kommandos oder einer Funktion einzugeben;
- gibt in der Bestimmung eines USING-Formats die Stellung des Dezimalpunktes an.

E

Wird benutzt, um in wissenschaftlicher Notation den Exponenten zu bestimmen (mit der "E"-Taste).

/

Divisionszeichen

[*] Multiplikationszeichen

- wird benutzt, um eine Feld-Variable in INPUT , PRINT etc. zu bestimmen.

[+] Additionszeichen

[-] Subtraktionszeichen

[SHIFT] [^] Wird benutzt, um Zahlen zu potenzieren und um das Exponenten-Anzeige-System für numerische Daten in USING-Anweisungen zu spezifizieren.

[SHIFT] [<] Werden benutzt, wenn logische Operatoren in einer
[SHIFT] [>] IF-Sequenz eingegeben werden sollen.

[DEF] Wenn eines der folgenden 18 Zeichen (A, S, D, F, G, H, J, K, L, Z, X, C, V, B, N, M, =, SPaCe) nach Betätigen der DEF-Taste gedrückt wird, wird das Programm mit dem entsprechenden Kennbuchstaben gestartet.

[A] bis [Z] Buchstabentasten

Diese Tasten sind Ihnen wahrscheinlich von einer gewöhnlichen Schreibmaschine her vertraut.

Auf einfachen Tastendruck erscheinen Großbuchstaben in der Anzeige. In den Kleinschriftmodus schalten Sie um, indem Sie [SHIFT] und [SML] nacheinander drücken.

SPC
[ ] Wird beim Eingeben von Zeichen oder Programmen benutzt, um einen Leerschritt zu produzieren.

[=] - In Zuweisungs-Statements: weist der links vom "=" stehenden Variablen die rechts stehenden Zeichen (Buchstabe oder Zahl) zu.

- Wird weiterhin benutzt, wenn in einer IF-Sequenz logische Operatoren eingegeben werden.

[SHIFT] !"=$ %&@ Zum Ansprechen der jeweiligen Zeichen:
" - zum Ansprechen und Löschen von Zeichen;
- zum Bestimmen von Kennbuchstaben;
= - im USING-Statement wird hiermit die Anweisung für numerische Inhalte gegeben;
$ - wird zur Zuweisung von String-Variablen benutzt;
% - im USING-Statement wird hiermit die Anweisung für Buchstabeinhalte gegeben;
- weist hexadezimale Zahlen aus;
@ - wird für den Reservespeicher benutzt, wenn die Reservetaste als Programmtaste eingesetzt ist;
!% - als Zeichenstring innerhalb " " zu verwenden.

[SHIFT] [?] Zur Eingabe von CLOAD?

[SHIFT] [:] Zur Trennung von zwei oder mehr Befehlen in einer Programmzeile

[.] Bezeichnet eine Pause zwischen zwei Gleichungen, zwischen Variablen oder Kommentaren.

[SHIFT] [:]
- Zur Ausführung von Multi-Display (Anzeige von zwei oder mehr Werten / Inhalten zur gleichen Zeit).
- Schafft eine Pause zwischen Instruktion und Variable.

[SHIFT] [(]
[SHIFT] [)] Zur Eingabe von Klammern

[▶]
- Bewegt den Cursor nach rechts (auf einmaligen Tastendruck Bewegung um eine Stelle; wird die Taste festgehalten, Dauerfunktion).
- Zur Ausführung von Playback-Anweisungen
- Löscht bei manueller Rechnung Fehlermeldungen.
- Wird benutzt, um nach Durchführung eines Easy Simulation Programms das versteckte Ergebnis anzuzeigen.

[◀]
- Bewegt den Cursor nach links (auf einmaligen Tastendruck Bewegung um eine Stelle; wird die Taste festgehalten, Dauerfunktion).
- Sonst wie unter der [▶]-Taste beschrieben.

[SHIFT] [INS] Fügt einen Leerschritt ein, dabei erscheint das Zeichen "␣". Dieser Leerschritt wird vor das Zeichen gesetzt, auf dem der Cursor steht.

[SHIFT] [DEL] Löscht den Inhalt der Stelle, auf der der Cursor steht.

[SHIFT] [π] Zur Eingabe von Pi (π)

[SHIFT] [Y] : Zur Berechnung der Quadratwurzel.

[ENTER]
- Zur Eingabe einer Programmzeile in den Computer.
- Wird beim Schreiben von Programmen benötigt.
- Bedingt manuelle Rechnung oder direkte Ausführung eines Befehls durch den Computer.
- Zum Neustart eines Programms, das zeitweise durch ein INPUT- oder PRINT-Befehl unterbrochen wurde, außerdem zur Durchführung eines BASIC- oder Easy Simulation Programms.

[SHIFT] [P↔NP] Stellt die Verbindung zum Drucker her bzw. unterbindet sie (sofern ein zusätzlicher Drucker an den PC 1260/1261 angeschlossen ist).

[SHIFT] [HELP]
- Zum Aufrufen der HELP-Funktion
- Zum Auflisten der Musterschreibweise jedes Befehls
- Zum Auflisten der ASCII-Zeichen Codes
- Hebt Blockaden durch Fehler auf und listet den Fehlertyp auf.

[SHIFT] [SML]
- Zum Anwählen und Aufheben des Kleinschriftmodus (schaltet die Anzeige SML an bzw. aus).
- Die SMALL-Anzeige erscheint, wenn [SHIFT] und [SML] nacheinander gedrückt werden. Wenn Sie nun die Tasten [A], [B] und [C] betätigen, werden a, b und c in der Anzeige ausgegeben. Durch erneutes Drücken von [SHIFT] und [SML] schalten Sie den Kleinschriftmodus wieder aus, die Anzeige SML verschwindet und es werden wieder Großbuchstaben in der Anzeige ausgegeben.

Die [↑] - und [↓] -Tasten haben die folgenden Funktionen, je nach angewähltem Modus und Status des Computers:

Modus	Status	[↑]	[↓]
RUN	Programmdurchführung		
	Programm vorübergehend unterbrochen, INPUT-Statement wird ausgeführt PRINT-Statement wird ausgeführt Unterbrechung	um die nächste Zeile durchführen zu lassen.	Diese Taste festhalten, um bereits abgearbeitete Zeilen zur Anzeige zu bringen.
	Fehlermeldung während der Programmausführung		um die fehlerhafte Zeile zur Anzeige zu bringen
	TRON Stadium	zur Fehlersuche	Diese Taste festhalten, um bereits abgearbeitete Zeilen zur Anzeige zu bringen.
	Anderes Stadium	um ein gerade ermitteltes Ergebnis zur Anzeige zu bringen.	wie links

Wenn der Modus von RUN nach PRO geändert wird, wird dies nicht angezeigt!

PRO	Programm vorübergehend unterbrochen	um die unterbrochene Zeile anzuzeigen	wie links
	Fehler	um die fehlerhafte Zeile anzuzeigen.	wie links
	Anderes Stadium	um die erste Zeile anzuzeigen.	um die letzte Zeile anzuzeigen.

(Wenn die Programmzeile oder das EASY Simulation Programm angezeigt werden:)

		um die nächste Programmzeile oder das Easy Simulation Programm anzuzeigen.	um die vorherige Programmzeile oder das Easy Simulation Programm anzuzeigen.

- In der Anzeige ist die [ENTER] -Taste das gleiche wie ein Leerschritt.
- Wenn 11 Minuten lang keine Taste betätigt wird, schaltet sich der Computer automatisch ab.

ANHANG F

TECHNISCHE DATEN

Modell:	PC 1260/1261 Taschencomputer
Prozessor:	8 Bit CMOS CPU
Programmiersprache:	BASIC
Speicherkapazität:	System-ROM 40 K Bytes RAM: System: ca. 600 Bytes Anwender: Fester Speicherbereich 208 Bytes (A-Z,A$-Z$) Programm-/Datenbereich PC 1260 3198 Bytes PC 1261 9342 Bytes Reservespeicher 48 Bytes
Stack:	Unterroutine: 10 Stacks Funktion: 16 Stacks FOR-NEXT: 5 Stacks Daten: 8 Stacks
Operatoren:	Addition, Subtraktion, Multiplikation, Division, trigonometrische und inverstrigonometische Funktionen, logarithmische und Exponentialfunktionen, Winkelumkehrfunktionen, Quadrat und Quadratwurzel, Vorzeichen, Absolut, Integral, Verhältnis-Operatoren, logische Operatoren.
Rechengenauigkeit:	10 Stellen (Mantisse) + 2 Stellen (Exponent)
Editiermöglichkeit:	Cursor rechts und links, Zeile auf- und abwärts, Zeicheneinfügung, Zeichenlöschung
Speicherschutz:	CMOS, batteriegestützt
Anzeige:	Zweizeilige 24-Stellen-Flüssigkristallanzeige mit 5x7-Punkt-Zeichen.
Tastatur:	52 Tasten, alphabetisch, numerisch, Sonderzeichen, Funktionen, numerische Schablone, definierbare Tastenfunktionen

Stromversorgung:	6.0 V DC-Lithium-Zellen Typ: CR-2032 x 2
Stromverbrauch:	6.0 V = DC 0.03 W Ca. 300 Stunden, wenn ständig 555555555 angezeigt wird bei einer Betriebstemperatur von 20 Grad Celsius. Diese Betriebszeit kann leicht schwanken, in Abhängigkeit von Arbeitsmethoden etc. * Bei einstündigem Betrieb pro Tag hält die Batterie ca. 4 Monate. Dies gilt, wenn die Betriebsstunde aus 10 Minuten Rechnung und 50 Minuten Anzeige besteht.
Betriebstemperatur:	0 Grad Celsius bis 40 Grad Celsius
Abmessungen:	135 (B) x 70 (T) x 9.5 (H) mm
Gewicht:	ca. 115 g (mit Batterien)
Zubehör:	Deckel, 2 eingebaute Lithium-Zellen, zwei Tastaturschablonen und ein Handbuch
Zusatzgeräte:	Drucker/Mikrocassettenrecorder (CE-125) Drucker/Cassetten-Interface (CE-126P) etc.

ANHANG G

DIE BENUTZUNG VON PROGRAMMEN, DIE FÜR DEN PC 1245 ODER PC 1250/1251 ENTWICKELT WURDEN

MODIFIKATIONEN VON PROGRAMMEN DER PC 1245 SERIE (PC 1245, PC 1250, PC 1251)

Der PC 1260/1261 benutzt grundsätzlich das gleiche BASIC wie die PC 1245 Serie, allerdings mit einigen Erweiterungen. Wenn Sie die Programme, die für die PC 1245 Serie entwickelt wurden, auf dem PC 1260/1261 benutzen wollen, müssen Sie folgende Änderungen vornehmen:

1. Multiplikation ohne das "*"-Zeichen
Bei der PC 1245 Serie kann der Operator für die Multiplikation (*) fortgelassen werden (z. B. AB anstatt A*B, DC statt D*C). Dies ist auf dem PC 1260/1261 nicht möglich, da dieser zwei aufeinanderfolgende Zeichen als einfache Variable behandelt. Benutzen Sie daher die Schreibweise rechts im Beispiel:

(z. B.) A = SIN BC ⟶ A = SIN (B*C)

2. Definition von Index-Variablen mit dem DIM-Befehl
Wenn auf den Computern der PC 1245 Serie der Befehl DIM A(30) ausgeführt wird, werden Speicherplätze von A(27) bis A(30) als Erweiterung eines Definitionsbereiches einer festen Variablen reserviert.

Auf dem PC 1260/1261 wird jedoch bei der Ausführung von DIM A(30) ein getrennter Speicherbereich für die Feld-Variablen A(0) bis A(30) und das Feld mit dem Namen A reserviert.

Wenn Sie Index-Variable als Erweiterung von Festvariablen definieren wollen, benutzen Sie bitte die rechts im Beispiel angegeben Schreibweise:

(z. B.) DIM A(30) ⟶ A(30) = 0

3. Datenein-/ausgabefehl für Dateien auf Cassette:
Bei der PC 1245 Serie wird mit dem Befehl PRINT#C der Inhalt der Variablen C und aller folgenden abgespeichert.

Beim PC 1260/1261 jedoch wird mit diesem Befehl lediglich der Inhalt der Variablen C auf Band gespeichert. Um den Inhalt einer bestimmten Variablen und aller ihr folgenden auf Band abzuspeichern, benutzen Sie bitte die Schreibweise rechts im Beispiel:

(z. B.) PRINT#A ⟶ PRINT#A*
INPUT#C ⟶ INPUT#C*

4. Wert einer Schleifenvariablen nach Beendigung einer FOR-NEXT-Schleife

Der Wert einer Schleifenvariablen, den Sie nach Ausführung einer FOR-NEXT-Schleife erhalten, unterscheidet sich beim PC 1260/1261 von dem bei der PC 1245 Serie. Wird der Wert einer Schleifenvariablen in einem Bedingungsausdruck benutzt, müssen Sie ihn beim PC 1260/1261 gegenüber der PC 1245 Serie um 1 erhöhen.

(z. B.)
```
10 FOR I=0 TO 10
50 NEXT I
60 IF I=10 THEN 100
```
Ändern Sie den Wert in Zeile 60 folgendermaßen:

```
60 IF I=11 THEN 110
```
(Auf dem PC 1260/1261 müssen Sie den Wert einer Schleifenvariablen um eins erhöhen. Die Anzahl der Schleifendurchläufe bleibt jedoch gleich.)

5. Exponentialzeichen "E"

Der PC 1260/1261 benutzt den Großbuchstaben "E" für die Darstellung der Exponentiation. Die folgenden Änderungen werden dadurch erforderlich:

```
A = 1.234 E 5  ------>  A =1.234E5
B = E 6        ------>  B =1E6
```

Wird ein Programm der PC 1245 Serie von Band in den PC 1260/1261 eingelesen, wird diese Veränderung automatisch vom PC 1260/1261 durchgeführt.)

6. Der Zeichen-Code der PC 1245 Serie unterscheidet sich geringfügig von dem des PC 1260/1261. Wenn die folgenden Codes durch die CHR$-Funktion zugewiesen werden, ändern Sie die Codes:

ASCII	PC 1245	PC 1260/1261
39 (&27)	Ü	´
91 (&5B)	√	[
93 (&5D)	π	]
96 (&60)	E	Leerschnitt
251 (&FB)	- (ERROR)	π
252 (&FC)	- (ERROR)	√

Anmerkung: Wie oben gezeigt, verfügt der PC 1260/1261 nicht über die Zeichen Ü und E.

ZUSÄTZLICHE MODIFIKATIONEN

1. Die Modelle PC 1245, PC 1250 und PC 1251 verfügen über eine Programmzeilenspanne von 1 - 999. Der PC 1260/1261 hat eine erweiterte Spanne von 1 - 65279. Aus diesem Grund beansprucht die Zeilennummer 3 Bytes im RAM (bei der PC 1245 Serie nur 2 Bytes.). Die Änderung wird automatisch ausgeführt, wenn Sie Programme von Cassette in den Computer einladen. Es besteht die Möglichkeit eines Speicherüberlaufs, wenn Sie zu lange Programme laden oder ausführen lassen (ERROR 6).
   Wenn eine Zeile etwa 80 Bytes umfaßt, kann durch diese Veränderung das Zeilenende gelöscht werden.

2. Wenn Sie ein Programm der PC 1245 Serie in den PC 1260/1261 einladen, wird aufgrund der Änderung in der Zeilennummerierung (von 2 auf 3 Bytes) die BUSY-Anzeige noch für einige Sekunden aufleuchten, nachdem das Band zu laufen aufgehört hat. Während dieser Zeit wird das "*"-Zeichen in der rechten unteren Ecke des Zeilendisplays sichtbar sein.

ÄNDERUNGEN BEZÜGLICH DES PC 1401

1. Änderungen der definierbaren ","-Taste

Bei dem PC 1260/1261 ist die ","-Taste keine definierbare Taste. Aus diesem Grund müssen Sie, wenn in einem Programm eine Taste definiert werden muß, eine andere wählen.

Beispiel:

PC 1401		PC 1260/1261
100 ",":	⟶	100 "=":

2. Andere Anmerkungen

Der PC 1401 verfügt über einige Funktionskommandos mehr als der PC 1260/1261. Darum wird bei der Programmausführung immer dann eine Fehlermeldung auftauchen, wenn ein Programm geschrieben oder eingelesen wird, das Befehle enthält, die der PC 1260/1261 nicht verarbeiten kann.

Wird ein Programm mit solchen Befehlen von Band eingelesen, werden diese Befehle durch "␣ " ersetzt und dann in der Anzeige ausgegeben.

SHARP CORPORATION

OSAKA, JAPAN

1984 © SHARP CORPORATION
Printed in Japan/Imprimé au Japon
4L3T(TINSG4175CCZZ)②